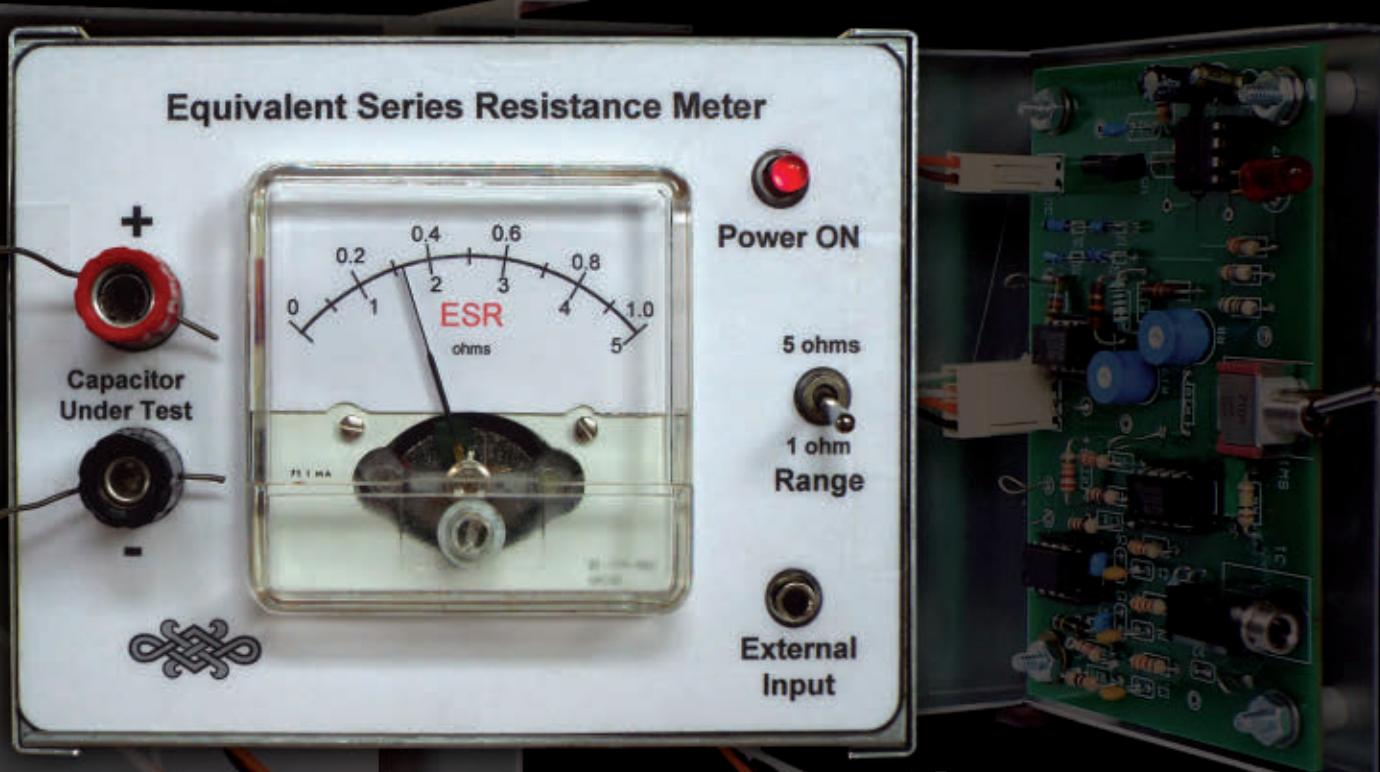


NUTS AND VOLTS

www.nutsvolts.com
January 2016

Are Your CAPACITORS suffering from Diminished Capacity?

Build This ESR Meter And Find Out!

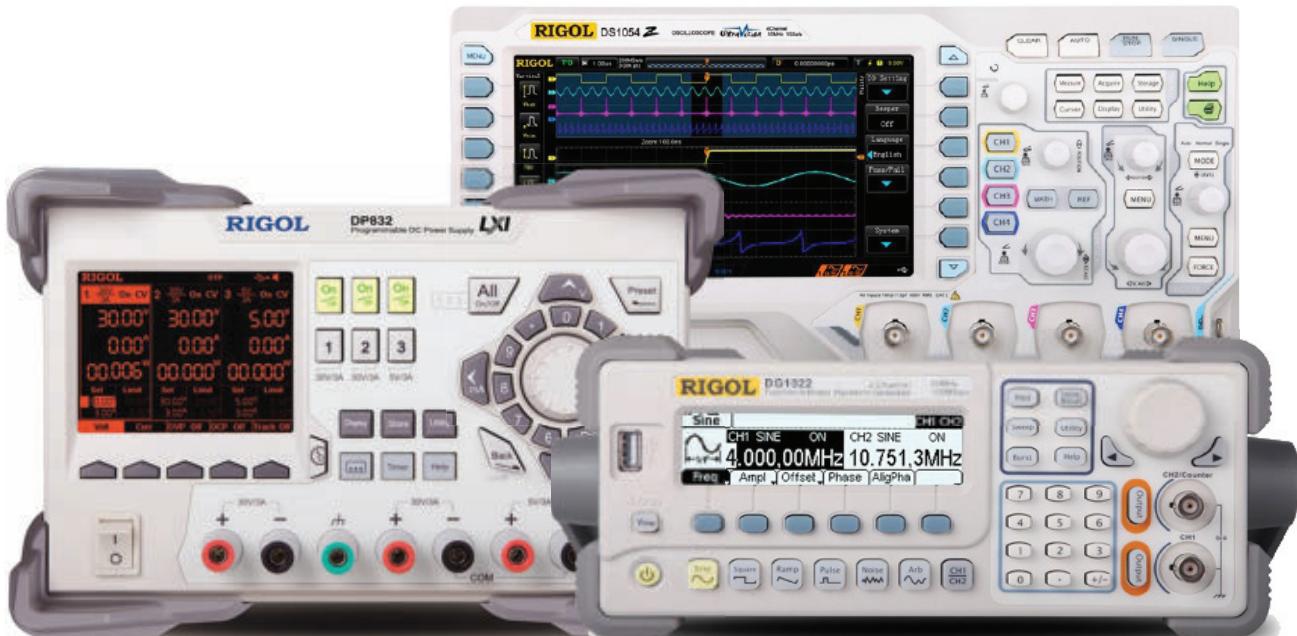


GET SMARTER with SmartThings and Zigbee
THE LOWDOWN on low power AVR
MAKE A KEYTAR from a Game Keyboard



The Future is Here

uncompromised performance, outstanding value



Rigol DS1054Z
50MHz - 4Ch Oscilloscope

\$399

Oscilloscopes



Rigol DP832
Triple Output Power Supply

\$450

Power Supplies



Rigol DG1022
20Mhz Function/Arb Generator

\$379

Generators



" Well designed, easy to use "

Ethan T.



" Tremendous Value "

Nicholas I.



" Hands Down the Best! "

Chris O.



Over 250,000 Satisfied Customers

Call your Sales Engineer for professional
solutions and helpful advice.

1-877-571-7901

FREE SHIPPING
on orders over \$99

Over 150 Brands
expert advice

Rigol 30 Day
Money Back Guarantee

HITEC CHARGE WITH CONFIDENCE

Experience the ease and reliability of battery charging with our dual port AC/DC charge units, the X2 AC Plus and X2 AC Pro! Designed with maximum efficient power distribution capability, these power-packed chargers provide flexible battery charging for all your chemistry types whether you are in the workshop or out in the field. Grab our optional WiFi module to configure your settings and keep an eye on your charge progress with your Smartphone or PC!

GET CONFIDENT WITH HITEC!



X2 AC PRO

- AC Mode Output: Total 200W Power with Power Distribution
- DC Mode Output: 200W Port A, 100W Port B for Total 300W Power
- Includes 60W Soldering Iron, 400 to 840°F

X2 AC PLUS

- AC Mode Output: Total 100W Power with Power Distribution
- DC Mode Maximum Output: 100W Per Port for Total 200W Power
- 10 Programmable Profile Settings

BOTH CHARGERS FEATURE:

- Power Distribution Mode for Ultimate Flexibility
 - Use Either 11-18V DC or 100-240V AC Power Sources
- Adjustable Charge Current Rate of 0.1 ~ 10.0 Amps Each Port
- USB Support for PC and Wi-Fi Smart Phone Interface

LiPo
1-6 cell

LiHV
1-6 cell

LiFe
1-6 cell

Lilon
1-6 cell

NiMH
1-15 cell

NiCd
1-15 cell

PB
2-20V

January 2016

08 Q&A

Reader Questions Answered Here

Topics this month include a scanner frequency change, a simple remote control, and a telephone off-hook indicator.

■ With Tim Brown

15 The Ham's Wireless Workbench

Practical Technology from the Ham World

Transmission Lines and SWR.

Learn about a basic element of transmission lines: the standing wave ratio. Find out why it's important and how to measure it.

■ By Ward Silver N0AX

24 Build an ESR Meter for Your Test Bench

Internal resistance within capacitors — known as Equivalent Series Resistance — can increase over time, causing power loss and/or heating. With this device, you can test your stash of capacitors so you can weed out any bad ones before you install them in your projects.

■ By Larry Coyle

30 Rock It with a MIDI Keytar

Using a Rock Band 3 keyboard, you can construct a very inexpensive, yet very capable MIDI controller for use in the studio or on stage.

■ By Dan Levin

35 SmartThings and the Device Maker

Discover a home automation platform with a cloud-based development environment that lets you create custom apps and microcontroller-based devices to talk to over Zigbee that you can control with your smartphone.

■ By John Rucker

45 Vintage Computing — Five Easy Projects for Your Mentor's Friend

Utilizing the retro computer described last month, here are some fun projects to get you started on a technology adventure, exploring computers and programming.

■ By Dane Weston

54 Beyond the Arduino — Part 8

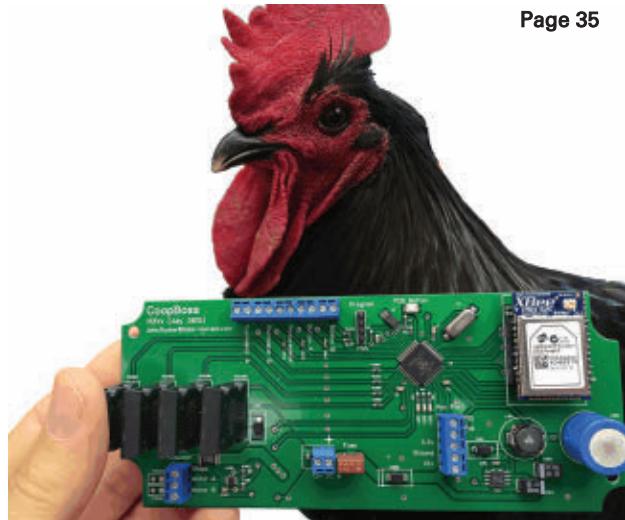
Power: Less is definitely more here as we focus on ways to reduce the power consumption in our projects so they can run for longer on less.

■ By Andrew Retallack

Page 30



Page 35



Departments

06	DEVELOPING PERSPECTIVES	44	ELECTRO-NET
	<i>Easing into STEM</i>	52	NV WEBSTORE
07	READER FEEDBACK	62	CLASSIFIEDS
20	NEW PRODUCTS	64	TECH FORUM
21	SHOWCASE	66	AD INDEX

Subscription Information			
Nuts & Volts — PO Box 15277			
North Hollywood, CA 91615-9218			
Call 877-525-2539 or go to www.nutsvolts.com			
Subscribe • Gift • Renewal • Change of Info			

Nuts & Volts (ISSN 1528-9885/CDN Pub Agree #40702530) is published monthly for \$26.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL MAILING OFFICES. POSTMASTER: Send address changes to **Nuts & Volts, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@nutsvolts.com.

Connecting industries through superior embedded technology for over 30 years



Single Board Computer



TS-7250-V2

1GHz ARM Computer with Customizable
FPGA-Driven PC/104 Connector
and Several Interfaces at Industrial Temp

Starting at
\$165 **\$199**
Qty 100 Qty 1

Computer on Module



TS-4900

Industrial High Performance
i.MX6 Module with Wireless
Connectivity and Flash Storage

Starting at
\$89 **\$122**
Qty 100 Qty 1

Touch Panel PC



TS-TPC-7990

7" High End i.MX6 Mountable
Panel PC with Dev Tools Such
as Debian GNU and QTCreator

Starting at
\$299 **\$342**
Qty 100 Qty 1





by Bryan Bergeron, Editor

DEVELOPING PERSPECTIVES

Easing Into STEM

I'm often asked what the best way is to support STEM (Science, Technology, Engineering, and Math) education with electronics. At the high school level, as soon as I start talking about Arduino boards and sensors, teachers tend to run away. It's intimidating to set up an electronics workshop from scratch. Think of all the necessary infrastructure that needs to be constructed — from multimeters and soldering irons to parts bins — and the components to fill them.

An alternative to a "made from scratch" approach is to use a kit or system that's been preconfigured with sensors and the tools to collect and display the data in real time. I've used TechBasic (\$15; ByeWorks.us) to turn my iPhone into a data collection platform.

I've gone as far as taping my phone to the spokes of

my bike during an off-road excursion. The concern, of course, was losing my phone. As I've discussed in previous editorials, TechBasic enables you to access the various sensors in the iPhone, display the data graphically, and massage the data as you see fit — all using a variant of the BASIC language. I've seen videos in which users tie their cell phones to kites and even solid fuel model rockets.

A way to get your hands on data without putting your phone at risk is to use a wireless sensor such as the PocketLab (\$98; thepocketlab.com). The 2.5" x 5/8" x 1-1/8" device is a BlueTooth-connected sensor cluster that collects data on temperature, barometric pressure, magnetic field, angular velocity position, and acceleration. The PocketLab is based on the TI CC2541, which I've used in the form of a fob-based evaluation kit available from Texas Instruments (TI). I found the hardware promising, but the software severely lacking. TechBasic provides support for the TI fob if you're into programming.

The folks at PocketLab also addressed the software problem, adding in support for cloud storage/sharing — the real advantage of this device over TechBasic. Not only are data displayed in real time, but they automatically move from the PocketLab to your Android or iOS device to the cloud, where data can be downloaded to your laptop for evaluation, manipulation, and analysis.

Also, while the TI fob is a bit clunky, the PocketLab's easy to handle plastic enclosure is mainly air, and the largest heaviest component by far is the coin battery. As an aside, PocketLab is one of those KickStarter success stories, raising \$100K in the time they had hoped for \$20K.

So, with environmental recorder in hand, what is one supposed to do to get all of this exposure to science, technology, engineering, and math? Well, as long as the experiment can be contained within the range of a Bluetooth device — say, in a classroom or on your person if you're outside — it's up to your imagination.

I wish I had access to a sensor-packed cell phone or an affordable wireless sensor package when I was studying Physics. I can still remember writing down rows of numbers from acceleration experiments. And forget about graphing results. That took hours.

So, in theory at least, with all the drudgery gone from doing science, everyone should be free to exercise their creativity, instead of spending time filling notebooks full of

Published Monthly By
T & L Publications, Inc.
 430 Princeland Ct.
 Corona, CA 92879-1300
(951) 371-8497
FAX (951) 371-3052
 Webstore orders only 1-800-783-4624
www.nutsvolts.com

Subscription Orders
 Toll Free **1-877-525-2539**
 Outside US **1-818-487-4545**
 P.O. Box 15277
 North Hollywood, CA 91615

FOUNDER
 Jack Lemieux

PUBLISHER
 Larry Lemieux
publisher@nutsvolts.com

**ASSOCIATE PUBLISHER/
 ADVERTISING SALES**
 Robin Lemieux
robin@nutsvolts.com

EDITOR
 Bryan Bergeron
techedit-nutsvolts@yahoo.com

VP OF OPERATIONS
 Vern Graner
vern@nutsvolts.com

CONTRIBUTING EDITORS
 Tim Brown Andrew Retallack
 Ward Silver

CONTRIBUTORS
 Dane Weston John Rucker
 Larry Coyle Dan Levin

CIRCULATION DEPARTMENT
subscribe@nutsvolts.com

SHOW COORDINATOR
 Audrey Lemieux

WEBSTORE MARKETING
 Brian Kirkpatrick
sales@nutsvolts.com

WEBSTORE MANAGER
 Sean Lemieux

ADMINISTRATIVE STAFF
 Re Gandara

Copyright © 2016 by T & L Publications, Inc.
 All Rights Reserved

All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *Nuts & Volts Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *Nuts & Volts*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *Nuts & Volts*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: 430 Princeland Court, Corona, CA 92879.

data. If you've used data collection and analysis as part of your STEM curriculum, please consider contributing to the reader forum so that other educators can learn from your experience. **NV**

READER FEEDBACK

Taking Time to Explain

Nuts & Volts and *SERVO* are unquestionably the best technical magazines available to the general public, and Bryan Bergeron's editorials are appropriately thought-provoking and informative. I would like to answer the three questions he posed at the very end of his Nov 2015 editorial as briefly as I can.

**ANSWER #1. MAGNETIZED
 MECHANICAL WATCHES DO NOT
 RUN FASTER BECAUSE OF
 DECREASED FRICTION.**

Vintage mechanical watches used steel hairsprings in their balance wheel assemblies as opposed to

more modern watches that converted to non-magnetic alloys (and prominently display those words on the dial as an inducement to the buyer). The moment of inertia of the balance wheel – restrained by the torsional spring rate of the spiral hairspring – constitutes a mechanical oscillator that will define the natural frequency of the combination, and hence the speed or rate of a watch.

These hairsprings are very fragile ... with small watches, their material thickness can be just a few thousandths of an inch, and the airgap in the spiral configuration must be maintained as the hairspring

Continued on page 62

Make up to \$100 an Hour or More!



Be an FCC LICENSED ELECTRONIC TECHNICIAN!

**Get your "FCC Commercial License" with our
 proven Home-Study Course!**

- No costly school. No classes to attend.
- Learn at home. Low cost!
- No previous experience needed!
- GUARANTEED PASS! You get
 your FCC License or money refunded!



**Your "ticket" to thousands of high paying jobs in Radio-TV,
 Communications, Avionics, Radar, Maritime and more.**

Call for FREE info kit: 800-877-8433 ext. 109

or, visit: www.LicenseTraining.com

COMMAND PRODUCTIONS • FCC License Training
 Industrial Center, 480 Gate Five Rd., PO Box 3000, Sausalito, CA 94966-3000



In this column, Tim answers questions about all aspects of electronics, including computer hardware, software, circuits, electronic theory, troubleshooting, and anything else of interest to the hobbyist. Feel free to participate with your questions, comments, or suggestions. **Send all questions and comments to: Q&A@nutsvolts.com.**

- **Scanner Frequency Change**
- **Simple Remote Control**
- **Telephone Off-Hook Indicator**
- **Mailbag**

Post comments on this article at www.nutsvolts.com/index.php?/magazine/article/january2016_QA.

Scanner Frequency Change

OI have a RadioShack Pro-7A Police and Fire scanner (eight-channel, crystal-controlled) that operates on 148 MHz to 174 MHz. Would it be possible to change crystals so that it can operate on the 144 MHz to 148 MHz two-meter ham band, or would other modifications be necessary?

— Gary Ross

A The Realistic Pro 7A is a vintage scanner which was marketed by RadioShack in the early to mid '70s (I found an ad in a 1973 magazine) which uses eight individual crystals to control its eight channels which are scanned until the radio finds an active transmission. I have included photos of the top and crystal holder portion of the Pro-7A in **Figures 1** and **2**.

The best I can understand about these early scanners is that they scan only one frequency per channel (there are a lot of different users in the 148 to 174 MHz band) unlike the modern scanners which scan entire frequency bands. To convert the Pro-7A, you start by deciding which frequencies you want to receive, such as a 145 MHz Listening Frequency (LF) in my example. Then, you have to calculate the Crystal Frequency (CF). This gets a little complicated, so hang on.

Most RadioShack (Realistic) crystal scanners used the third harmonic of the crystal to convert the incoming radio signal to the Intermediate Frequency (IF) and a 10.7 MHz IF (if your manual says your unit has a 10.8 MHz IF, change the number in the formula from 10.7 to 10.8). Use the formula:

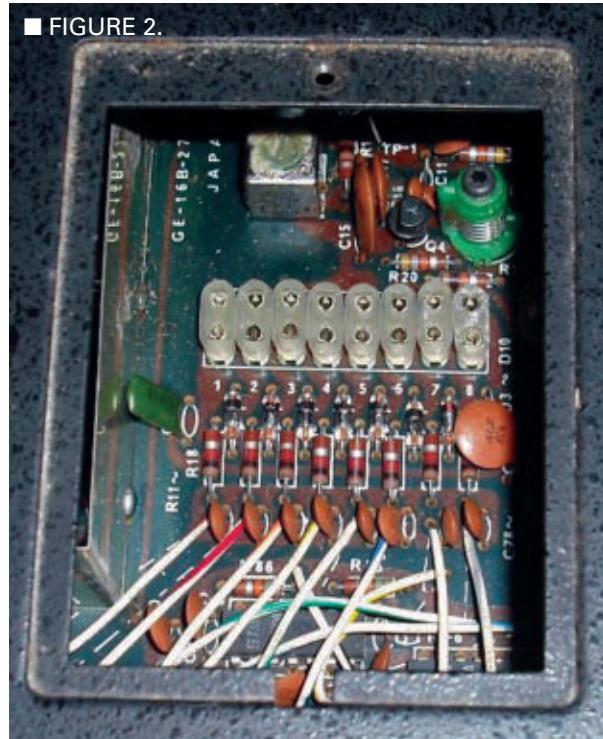
$$CF = (LF - IF)/3$$

For my example, the CF will be $(145 \text{ MHz} - 10.7 \text{ MHz})/3 = 44.767 \text{ MHz}$. Good vendors for crystals are Ken's Electronics, Digi-Key and Doug Baird (see Q&A **SIDELINES**). These crystals sell for between \$10 and \$15, so a full set of eight will cost between \$80 and \$120. A new scanner with 200 channels covering VHF Low (26-54 MHz), Aircraft (108-136.99166 MHz), Police/Fire (137-174 MHz), and UHF (380-512 MHz) will cost around \$120. Maybe we can get more information from our

■ FIGURE 1



■ FIGURE 2



readers who have converted a vintage scanner.

For those interested in scanners which can operate at emergency and aircraft frequencies, I have a base station scanner which can

be used as a stationary unit or installed in a vehicle. It has capabilities of 10 banks with 100 programmable channels per bank (1,000 channels, but you must switch banks manually), can receive AM, FM, or continuous tone signals, and can handle regular analog or digital broadcasts (many agencies are going to digital broadcasts which cannot be received by the older scanners). A very important addition to this scanner is the ability to access the 800 MHz channels being used in emergency situations to allow inter-agency communications.

This unit can also handle trunking, which is a radio version of networking in which a computer assigns one of many available frequencies to two communicating parties rather than having fixed frequencies for each agency. Trunking effectively gives agencies more bandwidth by allowing multiple conversations simultaneously without

Q&A SIDELINES

Scanner Frequency Change
Ken's Electronics Scanner Crystals
www.kenselectronics.com/lists/scanxtal.htm

Digi-Key Scanner Crystals
www.digikey.com/product-search/en/crystals-and-oscillators/crystals/852333?k=crystals

Doug Baird's Scanner Crystals
<http://dougbaird.net/crystalsforsale.html>

Simple Remote Control
NewEgg Item# 9SIA86V2Z33987 433 MHz Transmitter/Receiver Kits (Models MX-FS-03V/MX-05V) for Arduino and Raspberry Pi
www.newegg.com/Product/Product.aspx?Item=9SIA86V2Z33987&nm_mc=KNC-GoogleMKP-PC&cm_mmc=KNC-GoogleMKP-PC-pla-Gadgets-9SIA86V2Z33987&gclid=COM73vj69MYCFZIWHwodMqQMDA&gclsrc=aw.ds

Amazon SMAKN 422 MHz Transmitter/Receiver Kit for



■ FIGURE 3.

interfering with each other. This unit automatically centers its operating frequency to the received signal, which is a feature I always liked on the old analog "dial" radios where I could manually vary the tuning to achieve better reception. This unit also has USB capabilities, so you can download frequencies or trunk maps from your computer.

Simple Remote Control

OI am looking for a very simple small single channel remote control. I only need it to transmit about 25-50 feet or even less, but it does need to be RF and not infrared. I will need both the transmitter and receiver. A simple relay contact or open collector on the output of the receiver is all I need. Do you have any ideas on where to get such a device or perhaps a couple of circuits if I have to roll my own?

— Tom Bohacek

AI found remote control transmitter/receiver kits at NewEgg and Amazon which may meet your requirements (see Q&A SIDELINES). These are just two of many 433 MHz transmitter/receiver kits. I have included a block diagram of the transmitter/receiver combination connections to a control system in Figure 3. The TX control device can be a microcontroller or a simple device such as a switch, and the RX control device can be a microcontroller, a simple relay, or another device you wish to activate. If you want to use an Arduino microcontroller for this, I found a good tutorial online on how to use 315 MHz RF transmitter and receiver modules with Arduinos at [BuildCraft.com](http://www.buildcircuit.com/how-to-use-rf-module-with-arduino).

The 433 MHz radio frequency is in the Industrial, Scientific, and Medical (ISM) band which may be used by others such as amateur radio operators. If you are using a microcontroller unit with the TX/RX set, you could code the signals from the transmitter to the receiver to avoid having interfering signals with your control signal. If you are using the transmitter/receiver kits to operate directly without a microcontroller, you will not have protection from interfering signals.

The good news is that ISM signals are usually fairly short range, so interference will not be a problem. I hope this information solves your control needs. Our readers may have some other suggestions for transmitter/receiver combinations for simple control.

Telephone Off-Hook Indicator

OI have several four-line analog (RCA) telephones in my home and my wife's office (located on our property). We do not use the fourth line and I'd like to use it to signal her when I am on my VoIP work telephone (a separate, PC connected telephone headset). My VoIP headset puts out 5 VDC (guessing at about 5-10 mA) when off-hook, so I have a signal to work

with. What I don't know is what kind of a signal do I need to provide to my analog phone line input to make it show off-hook/busy?

— Jon Wilbert

AThe telephone has been around since the Scottish born American inventor, Alexander Graham Bell famously spoke to his assistant, Thomas A. Watson on March 10, 1876, "Mr. Watson. Come here. I want to see you."

MAILBAG

Re: Electronics Around Vehicle Fuel Tanks

#1 I always enjoy reading your knowledgeable and well-written Q&A section. I particularly appreciated your May 2015 tips on batteries. Your August issue on electronics and fuel tanks was very informative, but I question the accuracy of the Figure 2 schematic. It would seem that the potentiometer can't affect the Fuel Quantity Ammeter, as shown, or am I missing something? Maybe this is not an issue, since you warned the reader not to implement their own circuits. Good advice!

Arne Berg, Huntington Beach, CA

Mark, I am glad I could help you. Technology today is moving so rapidly it is sometimes hard to keep up with. So, the questions of our readers are helping me to stay on top of the rapidly changing field of electronics.

#2 I just read your description of the fuel pump and fuel level measurement systems and had to chime in here as I have fairly extensive experience working on these — especially the Bosch Jetronic systems widely used on most European cars since the mid 1960s. Now, granted the fuel systems I've dealt with have mostly been 1990s and earlier so some things may have evolved, but your description of the safety features has no basis in any reality I've ever seen. There is no external motor or explosion-proof pump housings, nor is the impeller in the main pump made of plastic. Injection pumps need to deliver more than 40 PSI to the fuel rail (and much higher in some systems) while lasting thousands of hours, so they use roller type pumps made of steel.

The motor is actually completely open and fully immersed in fuel which provides cooling and lubrication. Yes, the windings, commutator, brushes, and all are in a thin aluminum housing that is completely filled with gasoline flowing through the pump. This pump is located either within the fuel tank itself or remotely under the car, and a small impeller type pre-pump is submerged in the fuel — again with the motor and wiring — usually with exposed terminals completely submerged in fuel. The fuel sender is a similar deal; it is invariably mounted not in the neck but to the fuel pickup tube in a bung at the top of the tank, and consists of a float ball attached to a simple open frame potentiometer immersed directly in the fuel.

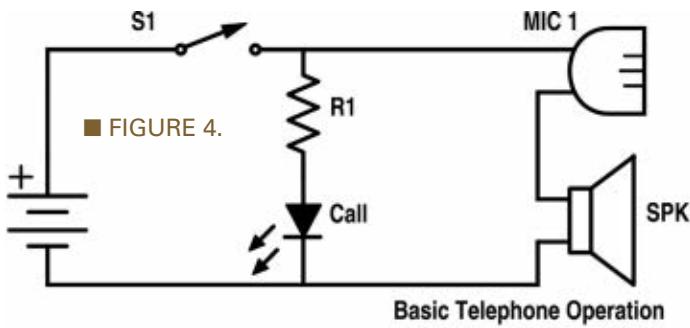
This all may sound crazy but that's how it's been done since the earliest fuel injected cars in the 1950s. It relies on the fact that fuel vapor in the tank displaces any oxygen that would otherwise be in there. Without oxygen, the fuel cannot burn. This is the same reason the common movie

scenario of a car's gas tank exploding is exceedingly rare in real life. If you want to see all this for yourself, search for "electric fuel pump teardown" on YouTube and you will find numerous videos showing all the gory details. I do agree with the basic advice that one should not attempt to install DIY devices within the fuel tank. It's just not worth the risk. Fuel systems can be dangerous to work on. The combination of pressurized gasoline, electricity, and potentially hot engine and exhaust components have caused more than a few car fires.

James Sweet

James, thanks for this information which is quite correct for the fuel system devices installed by the original manufacturer; the same standards are required for automotive aftermarket devices. These devices are extensively tested and certified as safe for use in automotive fuel systems before they are sold to consumers. My remarks and the ensuing warnings were directed at devices installed by electronics hobbyists who do not have the knowledge of how to make devices safe for such a hazardous application and who cannot conduct the extensive testing required. It kind of seems to defy logic, but electrical sparks in a liquid fuel will not trigger combustion. However, the same spark in the presence of fuel vapor will cause combustion. I am well aware of car fires because once I had an engine fire that erupted right after I had fueled the car and was in the station paying when we saw the smoke. The hood latch would not work and I was able to release the brakes, push in the clutch, and get the car away from the pumps before the fire fighters arrived on scene. (I don't know if I prevented a major disaster but I was trying to. I guess instead of being the hero of the day, I atoned for my own klutziness. In the past 30 or so years, I have never left off another oil cap.)

Automotive experts at our community college said it was a fuel leak onto the exhaust manifold, but when I checked out the car later, the oil cap had been left off the previous evening when I changed the oil. So, I think the fire was a result of oil pouring onto the hot manifold (a couple of weeks later, I had reworked the car's engine compartment and there were NO fuel leaks even after driving the car for a couple more years). The cashier at the station is probably still having therapy to return her eyes to the sockets. We both agree that our readers should be very careful when working around fuel systems or other potentially dangerous systems.

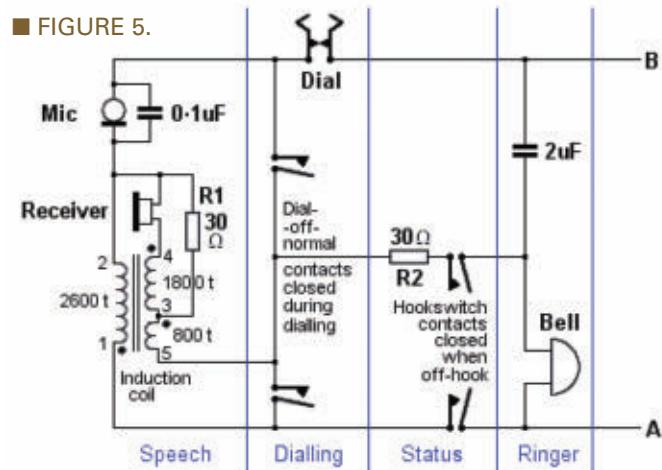


The basic telephone operation shown in **Figure 4** consists of a microphone and speaker connected together in series with a battery. In this "telephone," when switch S1 is closed, direct current (DC) can flow in the microphone and speaker which moves the speaker cone a bit but produces no sound other than an initial "pop." However, when someone speaks into the microphone (MIC 1), alternating current (AC) is produced which moves the speaker cone and "roughly" reproduces the sound spoken into the mic.

As an enhancement, I added an LED with a current-limiting resistor to show when a call was coming in. This is only a slight improvement over the old can and string acoustic phone, but a far cry from modern telephones which use computerized switching, satellite relays, terrestrial microwave links, wireless, cellular technology, and digital encoding.

The schematic for an old line powered phone is shown in **Figure 5**. This phone receives a 48 volt DC signal from the central switching office to power the phone. To ring the local phone, the central office sends an AC signal of 40 to 150 volts with a frequency of 20 to 40 Hz (cycles per second) superimposed on the 48 volt DC signal. When the receiver is lifted to answer a call, the ringer signal is disconnected and the line voltage between the local phone and the central office drops to eight volts DC (these voltages vary among the different carriers). [NOTE: At my home, the on-hook voltage is 52.4 VDC and the off-hook is 10.9 VDC.] To make a call, the off-hook voltage to the central office triggers a dial tone so the caller knows the call connection can be made. Most phones also include additional capacitors to cancel out the inductive reactance of the incoming phone lines.

The multi-line phone is just a normal telephone with multiple lines running to it. A four-line phone has four different lines run by the phone service provider and has the capabilities to switch between lines, put a line on hold, transfer a line, etc. An analog phone uses the amplitude modulation (amplitude of voltage on the line varies as the input voice signal increases) of the 48 volt DC line signal to transfer voice signals versus the digital phones which use digitally encoded signals to modulate the 48 volt DC signal.



The analogy between analog phone and AM radio signals is the radio has a high frequency local oscillator signal which is modulated by the input signal, whereas the phone modulates the DC signal. Thus, a four-line phone is essentially four individual phones in one body. I can imagine using several different frequency bands (frequency division multiplexing) on the two-wire pair to transmit several phone signals simultaneously with filters at the receiver end separating the different "lines," but I do not know of any phone service providers who do this on land lines for voice service to residential users.

FDM is used on long distance trunk lines, satellite communications, terrestrial microwave services, and broadband DSL. This could be a challenge for our readers: Design a multiple frequency/multi-line "phone" system for use in a home or office, but not connected to the phone company's lines (see Q&A **SIDELINES** for more details).

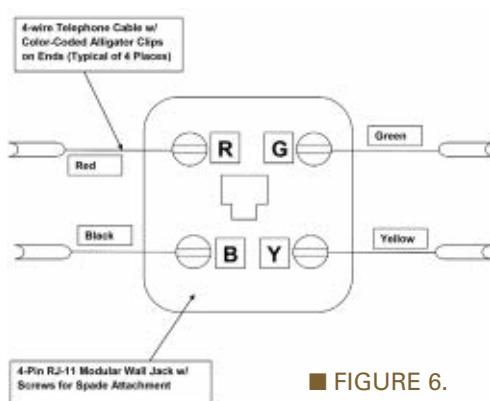
Now, let's answer your question on using an unused line to indicate that you are using your VoIP line. The easy way would be to hook your 5 VDC VoIP phone signal directly into the four-line phone's unused line to see if it worked. However, the easy way is not always the best way since you could damage either or both phones, plus the phone service providers require isolation of devices connected to their networks. First, use my homebrew test device shown in **Figure 6** to check the off-hook voltage on the four-line phone.

Hook the test device into the line you are going to test with a short piece of telephone cord with RJ-11 plugs on each end. Most four-line systems use the four-wire phone cable to provide conductors for two lines per cable (red-green on one line and yellow-black on the other), so you will have to determine which line is which to obtain the desired results. Hook a DVM to the red-green or yellow-black pairs, punch in the line you are testing on the four-line phone, take the phone off the hook, and check the off-hook voltage that your phone produces. Also check the off-hook voltage of your VoIP phone in the

same way. To prevent destroying either or both phones, use an optical isolator to prevent the current in one phone circuit from affecting the other phone circuit (it also prevents ground loops which can cause interference and will reduce noise transferred between the two systems). **Figure 7** shows an optical isolator circuit for connecting the VoIP phone line to the four-line phone's unused line.

The four input diodes insure correct operation of the circuit regardless of the phone line polarity (my residential line always shows as reversed on my store bought phone line tester). The opto-isolator prevents current from flowing between the VoIP phone system and the four-line system; this prevents damage to components in these circuits. If you do not want the unused four-line indicator to flash, eliminate the flash oscillator and tie both AND gate inputs to the opto-isolator output.

If your four-line indicator will operate from the five volt/11 mA TTL output from the AND gate, you can leave



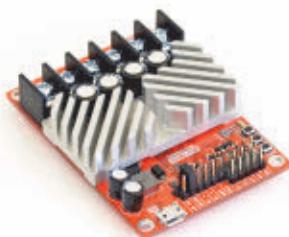
■ FIGURE 6.

out the op-amp (you could do this with your VoIP output but there would be NO isolation and no circuit protection). The op-amp section will require some calculation based on the output you need from the isolation circuit to drive the four-line indicator. A value of 1 KΩ for R_{in} is a good starting point; calculate R_f based on the needed output voltage from $V_{out}/V_{in} = (1+R_f/R_{in})$. Build the circuit first on a breadboard in case you need to modify (add and/or delete modules)

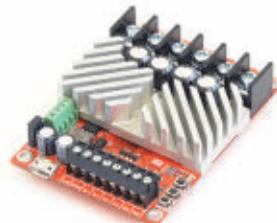
anything before you are ready to commit to a printed circuit board (PCB).

While writing this column, I experienced problems with my residential phone in which after a rain storm, the phone sounded like something was frying, our Internet signal was lost, and our phone/answering machine alternated between "Line in Use" (a.k.a., phone off-hook) and saying, "At the tone, leave a message." Our telco charges a large fee if their technician comes out and finds a problem with the internal residential wiring, so I checked

Motor Control



- 15 Amps Per Channel
- Dual Channel
- Quadrature Encoders
- DC Brushed Motors
- USB / RC / Serial



- 45 Amps Per Channel
- Dual Channel
- Quadrature Encoders
- DC Brushed Motors
- USB / RC / Serial



- 160 Amps Per Channel
- Dual Channel
- Quadrature Encoders
- DC Brushed Motors
- USB / RC / Serial

ION MOTION
CONTROL

www.ionmc.com

PARTS
EXPRESS
YOUR ELECTRONICS CONNECTION

OVER 18,000
ELECTRONIC PARTS IN STOCK



Visit us online to order a **FREE** copy of our sales flyer and to receive a special offer!

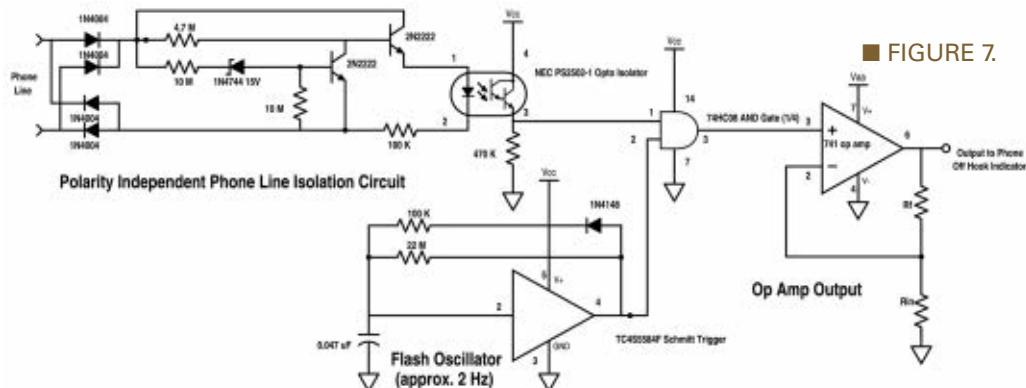
parts-express.com/nuts
1-800-338-0531

out the signal at the customer interface by removing the RJ-11 plug which connects the external phone wiring from the internal wiring (pretty messy after the 1-in-1,000 year storm).

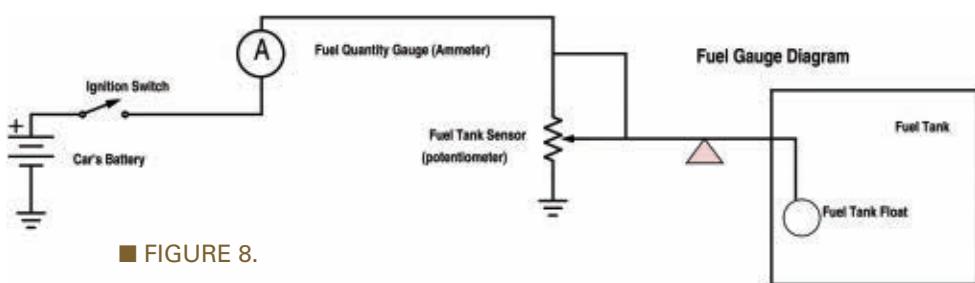
To see if the trouble was on the phone company side, I plugged in a phone that I have for testing phone systems and bingo! There was a plethora of noise on the telco lines. A spare working telephone is the best tool for testing your phone lines, plus they are cheap (mine was \$10).

A technician replaced the phone wire to the house and gave me his number to call if the problem returned the next time it rained. Well, it rained that very night and the next morning the phone problem was even worse. The phone

company bypassed a section of line about 0.1 miles from my house and the problem cleared up. **NV**



■ FIGURE 7.



■ FIGURE 8.



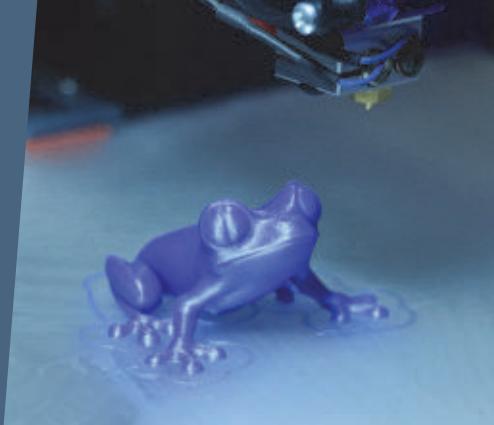
3D PRINTER KITS FROM MAKER'S TOOL WORKS

Minimax Kit

Full Kit \$699 | Motion Platform Kit \$399

PREMIUM QUALITY | ENTRY LEVEL PRICE

Print area : 200 x 230 x 200mm (8 x 9 x 8")
The same quality components as the MendelMax 3 in an entry level package.



Treefrog by MorenaP, <http://www.thingiverse.com/thing:18479>



Mendelmax 3 Kit

Full Kit \$1499 | Motion Platform Kit \$799

OUR FLAGSHIP | LARGER, FASTER, MORE VERSATILE

Dual Extruder Capable
Print area: 250 x 315 x 210mm (10 x 12.5 x 8")



Smooth linear motion | Steel and Aluminum structural components US made, open source control electronics

Premium E3D Metal hot End | Heated Bed

Supports 1.75 ABS/PLA & exotic filaments

Direct Drive Extruder | High Quality Components

MANUFACTURED IN THE USA

www.makerstoolworks.com

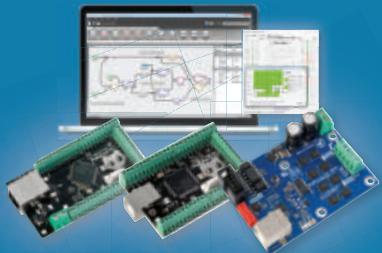
PoLabs

For more information or
software download please visit



www.poscope.com

Connect, Control PoKeys



USB: keyboard and joystick emulation

Ethernet: web server, cloud data storage,
Modbus TCP

I/O: digital I/O, 12-bit analog inputs, PWM
outputs

Counters: digital counters, encoder inputs

PLC: graphical programming with free
PoBlocks software

CNC: 1-8 axis motion controller (Mach3,
Mach4 or 3rd party app)

EASYSENSORS: I²C, 1-wire and analog
sensors

INTERACTION: matrix keyboard, matrix
LED, LCD

COMMUNICATION: free libraries for .NET,
ActiveX and cross-platform C/C++

Measure PoScope Mega1+



ALL IN ONE: Oscilloscope, Data Recorder,
Logic analyzer, Analog and digital signal
generator, Protocol Decoder

PROTOCOL decoding

SMALLEST USB 2.0 portable 2.5MS/s
oscilloscope

DATA ACQUISITION of analog and digital
signals

DATA RECORDING

EXPORT to CSV, XLS, PCM and HTML

SIMPLE USAGE of advanced features

LOWEST power consumption- less than 60mA

FREE SOFTWARE and updates

Nuts & Volts 12 CD-ROMs & Hat Special!



Price \$249.95

Plus FREE Shipping

Call 1 800 783-4624 or visit store.nutsvolts.com

High-Power Analog Autotuner



HF-AUTO

The HF-AUTO is a microprocessor controlled
fully automatic stand-alone tuner with a power
rating of 5 Watts to 1800 Watts that will work with
any transmitter built from the 1940s to the present.
HF Bands: 160m to 6m. Three antenna outputs:
SO239 coax. Dimensions: 12.5" W x 6.5" H x 16.5"
D. Weight: 25 lbs. (11.4 kg).



HF-AUTO-R Remote unit
Operate from 500' distance

 **PALSTAR**

Customer Support: 1-800-773-7931
Fax: (937) 773-8003

www.palstar.com

Transmission Lines and SWR

What's going on in that cable, anyway?

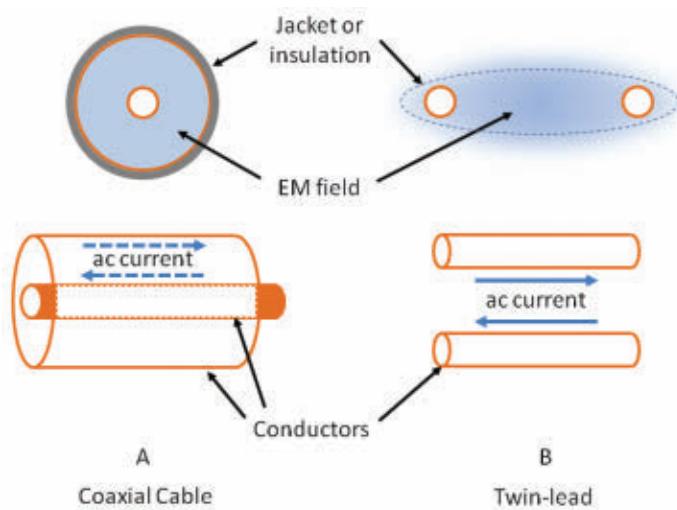
When shown an early radio station, the viewer was heard to remark, "Why do they call it wireless? I've never seen so many wires in all my life!" The fact that "wireless" requires a lot of wires is undeniable to those of us who have ever built or maintained any kind of radio facility — from a simple CB base station to a full-power broadcast facility. Most of the "wires" you'll find behind the equipment are more than simple strands, though. If they are conveying radio frequency (RF) signals from place to place, they are really transmission lines. In this article, we'll talk about a basic element of transmission lines — the standing wave ratio, or SWR — and why it's important and how to measure it.

Transmission Line Overview

Any conductor carrying an AC current can be treated as a transmission line, such as those overhead giants distributing AC utility power across the landscape. Incorporating all the different forms of transmission lines would fall considerably outside the scope of this article, so we'll limit the discussion to frequencies from about 1 MHz to 1 GHz, and to two common types of line: coaxial (or "coax") and parallel-conductor (a.k.a., open-wire, window line, ladder line, or twin-lead as we'll call it) as shown in **Figure 1**.

Current flows along the surface of the conductors (see the sidebar on "Skin Effect") in opposite

FIGURE 1. Coaxial cable (A) consists of a solid or stranded center conductor surrounded by an insulating plastic or air dielectric and a tubular shield that is either solid or woven wire braid. A plastic jacket surrounds the shield to protect the conductors. Twin-lead (B) consists of a pair of parallel solid or stranded wires. The wires are held in place by either molded plastic (window line, twin-lead) or by ceramic or plastic insulators (ladder line).



directions. Surprisingly, the RF energy flowing along the line doesn't really flow in the conductors where the current is. It travels as an electromagnetic (EM) wave in the space between and around the conductors. **Figure 1** indicates where the field is located in both coax and twin-lead. For coax, the field is completely contained within the dielectric between the center conductor and shield. For twin-lead, though, the field is strongest around and between the conductors but without a surrounding shield, some of the field extends into the space around the line.

This is why coax is so popular — it doesn't allow the signals inside to interact with signals and conductors outside the line. Twin-lead, on the other hand, has to be kept well away (a few line widths is sufficient) from other feed lines and any kind of metal surface. Why use twin-lead? It generally has lower losses than coax, so is a better choice when signal loss is an important consideration.

Characteristic Impedance

Both kinds of transmission lines are specified as having a characteristic impedance, represented by Z_0 . For example, popular RG-58 cable is designated to be a 50Ω cable, RG-6 is a 75Ω cable, and so on. If you measure the cable with an ohmmeter, you'll just get a reading of a few ohms. What's going on?

Z_0 applies to how EM waves flow through the cable, and it depends on the size of the conductors, the relative placement of the conductors, and the type of dielectric between them. (Formulas for Z_0 can be found online and in most RF engineering references.) Sometimes referred to as surge impedance, the characteristic impedance determines how the EM wave's energy is allocated between the electric and magnetic field as it travels along the cable.

You can experience acoustic characteristic impedance for yourself with a common soft drink straw and

a small diameter straw such as for mixed drinks. Blow a short puff of air through each and feel the resistance from the straw at the rising edge of the pulse. The larger straw allows more air to flow due to its lower impedance which you experience as back pressure resisting air flow. This analogy isn't exact but illustrates the general idea. If a short pulse of voltage is applied to a high Z_0 cable such as 300Ω twin-lead, the resulting current surge in the cable will be lower than for a low Z_0 cable such as 50Ω coax.

Characteristic impedance is important because of how energy gets into and out of a transmission line. As an EM wave travels — in space, along a wire, in a transmission line — any change in Z_0 causes some of the energy in the wave to be reflected in the opposite direction to the wave's travel. The bigger the difference in Z_0 , the more energy is reflected. Whether the new Z_0 is higher or lower determines the phase of the electric and magnetic fields of the reflected wave with respect to the forward wave.

Reading between the lines, so to speak, if an antenna or circuit or other load is attached to the line and has an impedance equal to Z_0 , all of the power traveling down the line will be transferred to whatever is attached. That's exactly what you want if you're transmitting a radio signal — for all of the transmitter's output to be transferred to the antenna where it is radiated away into space. Similarly, you want Z_0 to

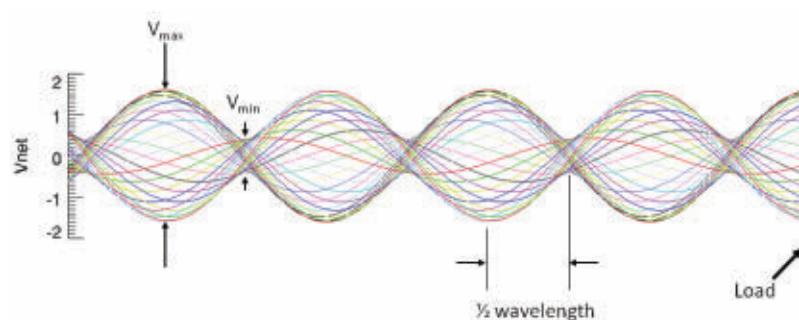


FIGURE 2. Standing waves on a transmission line, with the resulting voltages (V_{NET}) shown in different colors during one complete cycle of the applied voltage. The generated wave with an amplitude of 1 travels from left to right and is partially reflected by the load at right. $SWR = V_{MAX} / V_{MIN} = 1.6 / 0.4 = 4.0$.

(Graphic created by Wikipedia contributor, Interferometrist and is used under the Creative Commons Attribution-Share Alike 4.0 International license.)

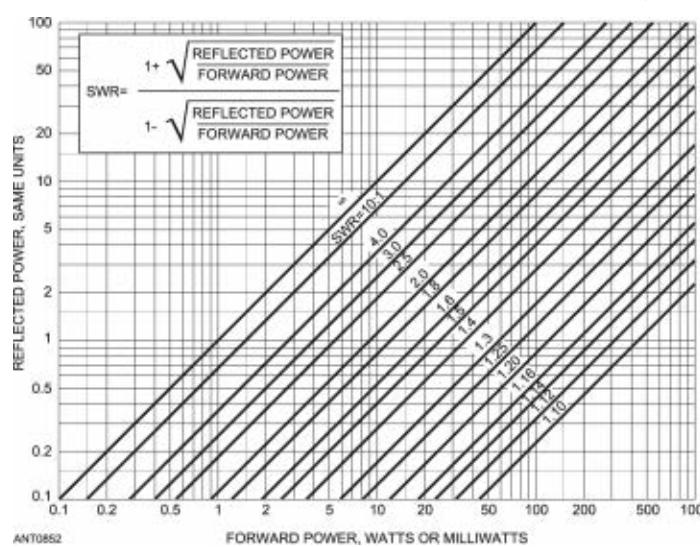


FIGURE 3. SWR as a function of forward and reflected power. (Graphic courtesy of the American Radio Relay League.)

match in the receiving direction so that all of that extremely weak signal picked up by the antenna is transferred to the line and thus carried to the receiver. (Transmission lines connected to antennas are usually just called "feed lines.")

Reflections on Standing Wave Ratio (SWR)

The condition in which the impedance of whatever is attached to the feed line equals Z_0 of the line is called matched. If the impedances are not equal, that's a mismatch. In most cases, Z_0 is not exactly

matched and so there are EM waves traveling up and down the cable, bouncing back and forth between the terminations at each end of the line. The device that applies power to the line is called the generator, and the device that takes power from the line is the load.

Forward refers to the direction from the generator to the load, and reflected refers to the opposite direction.

These waves set up an interference pattern that is stationary within the cable called standing waves. If all of the power is reflected at one end of the line, the pattern will include points at which the electric fields of the forward and reflected waves are out of phase and completely cancel — resulting in zero voltage. One-half wavelength away, the waves are in phase and add, thus doubling the voltage. **Figure 2** shows a calculated example of standing waves in which

the load impedance is four times higher than the line's, reflecting a portion of the forward wave. You can see a neat flame-based visualization of standing pressure waves in the YouTube video at

www.youtube.com/watch?v=6jfU74eNV_w. Think pressure = voltage and you've got it!

The ratio of the maximum and minimum voltages is the SWR. (If the SWR is calculated from the voltage, it is VSWR, and it can also be calculated from the standing waves of current as ISWR. The usual convention is to assume that references to SWR mean VSWR.) SWR is always equal to or greater

than 1, and is written as a ratio such as 1:1 or 1.5:1 or 3:1, and so forth.

SWR can also be calculated as the ratio of the feed line Z_0 and the load impedance — whichever is greater than 1. Since you already know the feed line's Z_0 and can measure the load's impedance, this is a lot more convenient than trying to measure maximum and minimum voltage inside the line:

$$SWR = \frac{V_{max}}{V_{min}}$$

$$SWR = \frac{Z_0}{Z_{load}} \text{ or } \frac{Z_{load}}{Z_0}$$

whichever is greater than 1

Another convenient way to measure SWR is by using forward power (P_f) traveling from the generator to the load and reflected power (P_r) traveling in the opposite direction:

$$SWR = \frac{1 + \sqrt{P_r / P_f}}{1 - \sqrt{P_r / P_f}}$$

Figure 3 shows a chart that converts any combination of forward and reflected power into an SWR value.

Effects of SWR (Radio and Non-Radio)

Why do we care so much about SWR? A clue was provided earlier in explaining that if the characteristic impedance of a line does not match the impedance of whatever load it is attached to, some of the power applied to the line is reflected at the load. Since we would like all of our expensive transmitter output power to be put to work as a radiated signal, it would be good to minimize reflected power at the load. SWR is



FIGURE 4. Inexpensive SWR meters (A) are useful up to about 30 MHz at power levels up to 100 watts. The more expensive meter (B) displays forward and reflected power simultaneously along with SWR on a crossed-needle display.

just a convenient way to measure the quality of the impedance match between our line and the load. The lower the SWR, the better the match!

Non-radio folks also care about SWR — especially with regard to high speed digital data. Data signals may “just” be voltage levels corresponding to 0 and 1, but it takes very high frequency components to make the sharp edges and narrow pulses our designs require.

A 100 Mbit/s data stream contains signal components in excess of 1 GHz! At those frequencies, every wire and PCB (printed circuit board) signal trace has to be treated as a type of parallel-conductor transmission line because it is.

If the Z_0 of a PCB trace does not match the output impedance of whatever circuit is generating the signal or the input impedance of whatever is receiving the signal, severe ringing, overshoot, undershoot, or multiple false transitions and glitches can occur. Search for application notes on “signal integrity” for detailed information about what PCB designers must do to control the data paths at today’s signaling rates.

Video signals — particularly analog video — can suffer from impedance mismatches, resulting in ghost images and distorted pictures.

Skin Effect

Above about 1 kHz, AC currents flow in an increasingly thin layer along the surface of conductors. This is the skin effect (https://en.wikipedia.org/wiki/Skin_effect). It occurs because eddy currents inside the conductor create magnetic fields that push current to the outer surface of the conductor. At 1 MHz in copper, most current is restricted to the conductor's outer 0.1 mm, and by 1 GHz, current is squeezed into a layer just a few μm thick.

The solution is to understand when transmission line considerations apply and terminate the signal traces in appropriately.

Measuring Impedance and SWR

To make accurate measurements at RF, instruments must be designed for that purpose. Low frequency multimeters simply can't be used. Nevertheless, inexpensive versions are available that don't cost a lot and still provide useful information. You just have to know where to shop!

SWR Meters

The most common transmission line instrument in a ham or CB station is the basic SWR meter shown in **Figure 4A**. The meter is used by setting the CAL control for a full-scale reading for forward power (FWD), then switching to the reflected (REF) position to read SWR. The meter sensitivity varies with frequency, requiring readjustment when using different bands; accuracy is low compared to a lab instrument. Available online and from CB shops for under \$30, these meters give a good idea whether SWR is high or not, and are useful in monitoring output power while operating or when adjusting an antenna.

More advanced meters like the Daiwa CN-101 in **Figure 4B** provide simultaneous power and SWR measurements with a crossed-needle meter. The unit displays both forward and reflected power with



FIGURE 5. The author's Bird Model 43 has been in use for a long time, but still provides reliable readings of forward and reverse power using a collection of elements that span 2 to 1,000 MHz.

independent meter movements. Curves on the meter face indicate SWR at the point where the needles cross. (The curves are the meter equivalent of the chart in **Figure 3** or the equation for SWR in terms of forward and reflected power.)

Accuracy of this type of instrument — typically costing around \$100 — is quite a bit better, no calibration adjustments are required over the specified frequency range, and the meters can handle the full amateur legal power of 1.5 kW in several ranges. Similar units are available for VHF and UHF operation.

These meters work by coupling a

small amount of power from the main feed line into a diode detector circuit. One circuit senses forward power and another senses reflected power — each driving an analog meter. The meters and SWR values are calibrated based on typical diode characteristics, and accuracies of a few percent are about as good as can be expected.

This is not up to lab standards but good enough for day-to-day operation, and repeatable enough to compare antennas. Digital

meters are also available which digitize the detector output signals and compute SWR for display as a numeric value.

Directional RF Wattmeters

A step up in accuracy from the SWR meters, directional RF wattmeters — such as my Bird Model 43 shown in **Figure 5** — measure forward and reflected power independently. A sensing element is inserted into the front of the meter and rotated according to the arrow to read power in the desired



FIGURE 6. The SARK-110 Antenna Analyzer is a miniature vector impedance meter that displays several types of measurements from 0.1 to 230 MHz. Data can be saved for future analysis on a PC, as well. (Photo courtesy of SteppIR.com.)

Similarities in Wave Behavior

This column only touches on how transmission lines actually work in order to discuss some of the common instruments you'll find in the ham station for working with impedance mismatches. A much better presentation is available from the AT&T Tech Channel at www.youtube.com/watch?v=Dovun0xIY1k. Called "Similarities in Wave Behavior," the tutorial uses a clever mechanical simulator called the Shive Wave Machine to show just how waves move in transmission lines, including reflection and standing waves. Some simple demonstrations illustrate the idea of SWR and what it measures. If transmission lines seem confusing, watch this video as a great place to start learning.

direction. The meter's user can then convert the readings into SWR, if desired.

Most hams and technicians only use SWR as an indication of whether excess reflected power is present, so an accurate display of reflected power serves the same purpose as an SWR value.

Sensing elements are calibrated for different power levels and frequency ranges. The list of Bird elements — also known as "slugs" — is available at www.birdrf.com via the Model 43 product sheet. Elements are available at frequencies from 2 to 1,000 MHz, and maximum power levels to 5,000W. Wide-range meters with digital displays are also available that perform the same functions automatically using a microprocessor.

Measuring Impedance Directly

Until relatively recently, making an accurate measurement of impedance at RF required some serious lab instrumentation. Impedance measurements vary with frequency, and can be upset by "stray" capacitance and even lead length of the measuring probe. Based on microprocessor technology developed for the mobile phone,

Great Lengths on Lines

Transmission lines are one of the most written-about topics in all of ham radio. More than just ways of carrying RF energy from place to place, they can act as filters and impedance transformers, too. Who knew that a simple "wire" could do all these tricks? You can learn a lot more about them via the ARRL's Technical Information Service's web page at www.arrl.org/transmission-lines.

FIGURE 7. The DG8SAQ Vector Network Analyzer not only measures impedance through 1.2 GHz but also makes gain, loss, and phase measurements of circuits and transmission lines. A laptop or other host PC is required to process the measurements taken by the unit shown in the photo. (Photo courtesy of SDRKits.com.)



manufacturers are now supplying impedance meters that provide excellent accuracy, record data for later analysis, and display the measurements in numerous formats.

The simplest type is called a "one-port vector impedance meter" as shown in **Figure 6**. This handheld instrument is about the size of a smartphone, costs under \$500, and has a color display that can show the data as impedance, SWR, or in other formats across several decades of frequency. The word "vector" signifies that the meter measures both magnitude and phase of the unknown impedance. These pocket-sized units are especially handy for field use.

A more capable instrument is the two-port vector network analyzer, or VNA shown in **Figure 7**. Where the one-port model can only measure whatever is attached at the single connector, the VNA can generate a signal, pass it through a circuit or cable, and measure how it is amplified, attenuated, phase-shifted, etc. VNAs used to be found only in engineering and research facilities, and cost \$10,000 or more. Now, you can buy units specified for use through 1 GHz for under \$1,000. These units are designed to be used on the workbench with a host PC.

Once you have the ability to measure impedance and power in a transmission line, many powerful

techniques are available to turn the data into an understanding of what's going on "in there." In turn, this allows you to design and troubleshoot your antenna or data transmission system. Transmission lines become a lot easier to work with once you have the ability to make the right measurements!

Correcting for SWR

Finally, what can you do about SWR if you have too much of it? That's a loaded question since how

much is too much depends on the application. Nevertheless, in the next column we'll take a look at techniques used by hams to reduce SWR by transforming and matching impedances across a wide range of frequencies. **NV**

Mike Ellithorp KF6OBI pointed out the URL given for the Mackie Company information has changed. You can find the CompactMixer_RefGuide at <https://supportloudtech.netx.net/loud-public/#/asset/6256>. Look on page 201 for the "Grounds, Shields, Hums, and Buzzes" material.

Premium tools for makers and creators

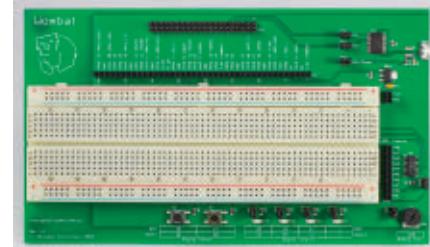
Learn PIC programming



Tutorials and training board

Clear lessons
C and assembler
Hands on examples
Includes enhanced mid-range PICs

Create original projects for Raspberry Pi



The Wombat prototyping board

Provides **easy access** to all GPIO pins
Adds **functionality** to your Pi, such as:
analog inputs, LEDs, pushbuttons,
USB serial console port
Includes a set of projects to get you started.

NEW! See our Raspberry Pi display add-ons

www.gooligum.com

NEW PRODUCTS

- HARDWARE
- SOFTWARE
- GADGETS
- TOOLS

RE-MASTERED PARKMASTER PRO

Building on the international success of their original ParkMaster 3D, Hitec's ParkMaster PRO is capable of the most extreme park flyer aerobatics. Square-section carbon fiber spars and longerons produce an unbeatably rigid airframe, while also shaving precious weight. The reinforced motor mount is suitable for more powerful motors, and an optional 4 mm carbon firewall plate offers increased stiffness, lower weight, and high tech bling.

The ParkMaster PRO is optimized for outdoor aerobatics and docile handling. It excels in confined spaces and moderate wind, and is easily capable of torque-rolls, rolling loops, and more. It is available as a kit or a Kit Plus version which includes the brushless motor, speed control, propeller with driver, and Karbonite servos. Pricing is as follows:

- ParkMaster Pro Kit Version; Est. Street Price: \$103.99
- ParkMaster Pro Kit Plus Version; Est. Street Price: \$205.99. Includes: HIMAX C 2816-1220 motor, MULTicont BL-30 S-BEC speed controller, 10 x 4.7" propeller with adapter, and four Nano Karbonite servos.
- Optional Carbon Firewall; Est. Street Price: \$23.99



For more information, contact:
Hitec RCD USA, Inc.
www.hitecrcd.com

PROFESSIONAL DIGITAL GEIGER COUNTER

Images Scientific Instruments is now offering a professional digital Geiger Counter that is NRC certification ready, which certifies that the radiation reading from the Geiger counter is accurate. This



certification can only be given by a US government licensed laboratory.

The GCA-07 DL pro Geiger counter detects alpha particles above 3 MeV in energy. The GM tube has a thin mica window that allows alpha radiation to be detected. It also detects beta radiation above 50 KeV, and X-ray and Gamma radiation above 7 KeV.

Counting resolution and range is 1 count per minute (CPM); 10,000 counts per second (CPS).

Radiation resolution and range is 0.001 mR/hr resolution, 1,000 mR/hr range (Imperial measurements); 0.01 uSv/hr resolution - 10 mSv/hr range (Metric).

The included data logger can record and playback over eight hours of radiation data. Retail price is \$545.95.

For more information, contact:
Images Scientific Instruments
www.imagesco.com

ENHANCED EFFECTS LIGHT w/ SILICONE BI-PIN LAMP

J2 LED Lighting, LLC is now offering a very different type of light socket cable: the Enhanced Effects Light, or EEL.

The EEL's innovation is in its ability to be mounted into High Density Urethane (HDU) styling and modeling board, which is a commonly used substrate material for industrial models, props, scenery, and signage. The EEL is targeted for





QKITS LTD
sales@qkits.com

Arduino • Raspberry Pi

Power Supplies
MG Chemicals
RFID

\$8.50
flat rate shipping

1 888 GO 4 KITS
Visit us at: www.qkits.com

NATIONAL RF, INC.

TYPE  75-NS-3

MINI HF RADIO

The 75-NS-3 covers 3.0 to 12 MHz and is offered as a semi-kit.



CUSTOMER PUTS IT IN A MEAT CAN!!
Visit www.NationalRF.com for this and other Radio Products!
Office: 858-565-1319

NKC electronics

MDE8051 Trainer
by Digilent

NKCelectronics.com/MDE8051

Includes the MDE8051 training board, power supply, serial cable

Purchase Orders are accepted from Educational Institutions, US Government and Research Centers

specific use in HDU or other rigid foam board material, and is excellent for waterjet cutting and CNC milling.

The EEL's cable socket fits into a mounting hole from the material's front side. The tapered fit of the socket's body locks it in place, and a front 5/8" diameter flange provides an insertion stop. The EEL cable socket light assembly is optimized for six pound density foam board when an 11 mm (7/16") hole is used, though it can be mounted into higher density boards with the hole size increased slightly to reduce insertion pressure, or with lower density foam board reducing the hole diameter for a tighter grip.

In addition to the cable socket, the EEL has an LED light source: the SBL (Silicone Bi-Pin Lamp). The SBL is a mid-power, 1.1 watt nominal, 110 lumen, warm white source that operates on 12 volts DC. The power consumption is low enough that several of the lights can be operated from a small 12 volt battery pack. The EEL greatly expands the lighting options available to the designer who uses high density foam board or is planning a new project and needs a unique creative means of combining light with a versatile substrate.

For more information, contact:
J2 LED Lighting, LLC
www.j2ledlighting.com

HIGH PERFORMANCE SMD SOCKET ADAPTER

Ironwood Electronics' new high performance SMD Socket Adapter (LS-BGA84H-61) allows users to socket a BGA chip having only the SMT pads on the target printed circuit board (PCB) without any





Explore
CircuitGizmos.com

Fast and simple shield interfacing with CGMicroBoard - No compiling!



Interface to Touch LCD
Micro SD card
Shield Compatible
USB interface
IR Tx/Rx, 1-wire, Analog
Multi serial/SPI/I2C
9-12V in, 5V/3.3V reg

ROBOT POWER *Extreme Motor Speed Control!*

MegaMoto GT- Mega Motor Control for Arduino™

- ♦ 6V-40V - 35A / 50A+ peak
- ♦ Single H-bridge or dual half
- ♦ Current and Temp protected
- ♦ Jumper select Enable & PWM
- ♦ Robust power terminals



The VYPER™

- ♦ 8V-42V - 125A / 200A peak!!
- ♦ Single H-bridge
- ♦ RC and Pot control
- ♦ Overload protected
- ♦ For your BIG bots

www.robotpower.com
Phone: 360-515-0691 • sales@robotpower.com

additional space for a socket. The companion SF-BGA84H-B-64 is attached directly to the target PCB using the same soldering methods as attaching a BGA IC. The target IC is soldered to the LS-BGA84H-61. The two parts are interconnected together with gold plated machined pins for high reliability. The height of the socket not including the IC is then 4.5 mm. This patented adapter set allows easy prototyping or upgrading of DDR memory chips reliably and inexpensively. Pricing for the LS-BGA84H-61 socket adapter is \$23 each.

For more information, contact:
Ironwood Electronics
www.ironwoodelectronics.com

COCK FIGHT – NEW EHX PEDAL

The EHX Cock Fight now available from Electro-Harmonix features two expressive filters with tunable frequencies to produce classic cocked wah and cocked talking pedal sounds – sans the rocking wah pedal. A built-in



Continued on page 66

The Nuts & Volts

Workbench Design Challenge Redux

The challenge is over!

All the entries have been reviewed and the judges have made their decisions.

The Winners Are:

Joe Fulton - Oscium iMSO-204L mixed signal oscilloscope

Nelson Moye - Parallax Starter Kit

Dan Lavin - BitScope Micro with Rasp/Pi Test & Measurement workstation

Dave Hunter - \$50 gift certificate to the Saelig online store

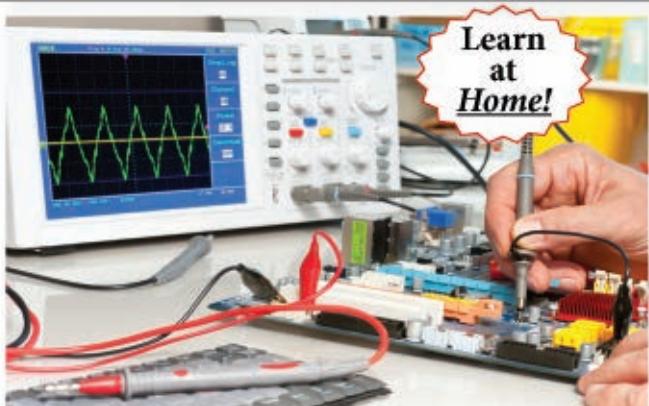
Cory McGinnis - Measurement Computing BTH-1208LS Wireless DAQ

Joe Menard - Oscium WiPry-Pro 2.4 GHz spectrum analyzer

Our thanks go out to the great companies that donated prizes for this challenge.

Parallax, Measurement Computing, Bitscope, Saelig, and Oscium!

See the results online at www.nutsvolts.com/the-workbench-challenge-2015-redux-winners



Electronics Courses

Cleveland Institute of Electronics

Train at home to be a professional electronics or computer technician! Learn with hands-on labs, instructor support, online exams, videos and instructor led chat rooms.

FREE course catalog www.cie-wc.edu or (800) 243-6446

- **NEW!** Robotics Automation Lab
- Industrial Electronics with PLC
- Electronics with FCC
- Electronics Troubleshooting
- **NEW!** Computer Security Specialist
- Broadcast Engineering
- PC Troubleshooting
- Networking

Visit www.cie-wc.edu and start today!

CIE: 1776 E. 17th St., Cleveland, OH 44114 • (800) 243-6446 • Registration 70-11-0002H



AP CIRCUITS
PCB Fabrication Since 1984

As low as...

\$9.95

each!

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com





5% Educational Discount

Introducing The New X Series!



- Large 8 inch color LCD
- 100MHz, 200MHz, bandwidth models
- Real-time sampling rate up to 1GSa/s
- Record length of 14Mpts, Waveform capture rate up to 60,000 wfs/s
- New generation of SPO technology
- Supports 256-level intensity grading and color temperature display
- SPO Technology means low noise

SDS 1000X Series Super Phosphor Oscilloscope



- Dual-channel, 120MHz maximum bandwidth, 20Vpp maximum output amplitude, high fidelity output with 80dB dynamic range.
- High-performance sampling system with 1.2GSa/s sampling rate and 16-bit, not 14-bit vertical resolution. No detail in your waveforms will be lost.
- Sweep and Burst function
- High precision Frequency Counter
- High fidelity, low jitter, 4.3" touch-screen display

SDG 2000X Series Function/Arbitrary Waveform Generator



- 3 independent controlled and isolated outputs, 32V/3.2A×2, 2.5V/3.3V/5V/3.2A×1, total 220W
- 5 digits Voltage, 4 digits Current Display, Minimum Resolution: 1mV/1mA
- Supports panel timing output functions
- 4.3 inch true color TFT- LCD 480x272 pixel display

SPD 3300X Series Programmable DC Power Supply

Best
Prices



585-385-1750
www.Saelig.com

Excellent Customer Service, Free Technical Support

Build an ESR Meter for YOUR TEST BENCH

By Larry Coyle

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?/magazine/article/january2016_ESRMeterforTestBench.



Time marches on, and eventually everything goes downhill. That includes me, you, and, surprisingly, most of those capacitors you've been hoarding in your junk box for years, just waiting for a project to put them to use. Why mention capacitors? Because high capacitance types like aluminum electrolytics and tantalums can slowly deteriorate over time. The internal resistance, called "Equivalent Series Resistance" (or ESR) can increase, causing power loss and heating. This can happen if the capacitor has been subjected to electrical stress or elevated temperature, or even while it's just sitting around in storage, not connected to anything.

With the instrument I describe in this article, you can test your store of capacitors or those in some vintage equipment you may be restoring, so you can weed out the ones that may not be up to par. Moreover, this design is easy to build and set up, using only common through-hole parts (no surface-mount devices!) and no microprocessors. In concert with this "retro" approach, the measurement result is displayed on a conventional moving coil panel meter.

I find this device to be a useful gadget to have around my work bench. I have a bunch of capacitors I have accumulated over many years — some of which have been salvaged from old equipment or cycled through several projects. There's no telling what abuse and degradation they may have suffered, and I definitely don't want to use a component in my next project that's going to let me down, no matter how pristine its appearance.

Measuring ESR

As detailed in the **sidebar** ("What a Capacitor Really Looks Like"), a number of factors contribute to power loss in a capacitor. These losses can be lumped together as ESR, which looks like a small resistance in series with an ideal (lossless) capacitor.

A simple technique for measuring ESR is to supply the capacitor with a known AC current (I_{cap}) at some frequency where the reactance of the capacitor is very low so that the ESR dominates. Measure the resulting AC voltage developed across the capacitor's terminals (V_{cap}) and you can find the ESR by dragging out Ohm's Law:

$$ESR = V_{cap}/I_{cap}$$

This is the basis of the ESR meter I describe in this article. A glance at the equivalent circuit model shown in the sidebar should make this clear.

All capacitors have an inductive component which can possibly interfere with the ESR measurement. In some ESR meters, a square wave or pulsed source is used to test the capacitor, and the resulting inductive spikes can cause an abnormally high ESR reading. Accordingly, I have incorporated a sine wave source into the design to avoid this possibility.

The block diagram in **Figure 1** shows that the ESR meter is made up of four fundamental sections: 1) a sinusoidal oscillator to supply an AC current to the capacitor under test; 2) an ESR detector to sense the AC voltage developed across the capacitor; 3) a meter amplifier and rectifier to display the ESR on a panel meter; and 4) a power converter and voltage regulator section similar to that found in many electronics assemblies.

The full electrical schematic diagram of the

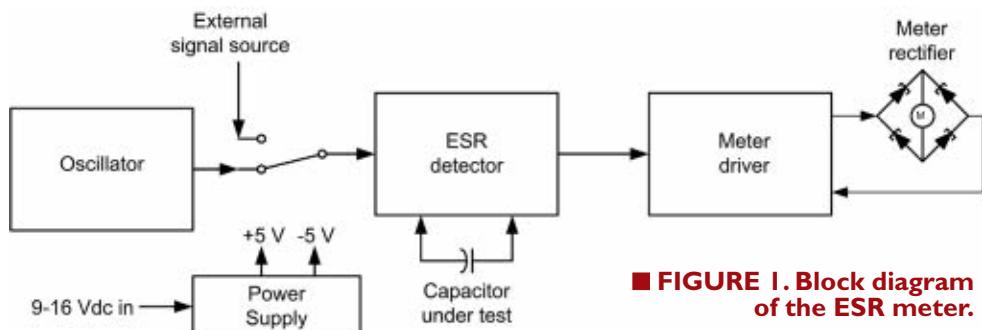


FIGURE 1. Block diagram of the ESR meter.

ESR meter is shown in **Figure 2**.

The Oscillator

This supplies the necessary AC signal for driving current through the capacitor being tested. The circuit here runs at approximately 100 kHz, which is an industry standard for making ESR measurements. One section of dual op-amp U1 functions as a phase-shift oscillator in this application. I like this circuit and have used it in several projects. It's simple to implement and gives a pretty good approximation of a sine wave. It's ideal for generating a fixed-frequency signal through audio frequencies and beyond, if the requirements are not too demanding.

The other section of U1 acts as a buffer and amplifier.

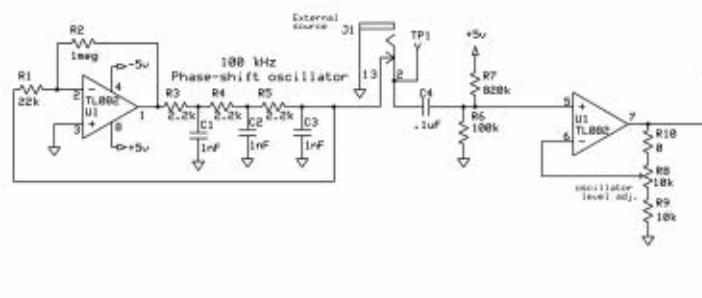
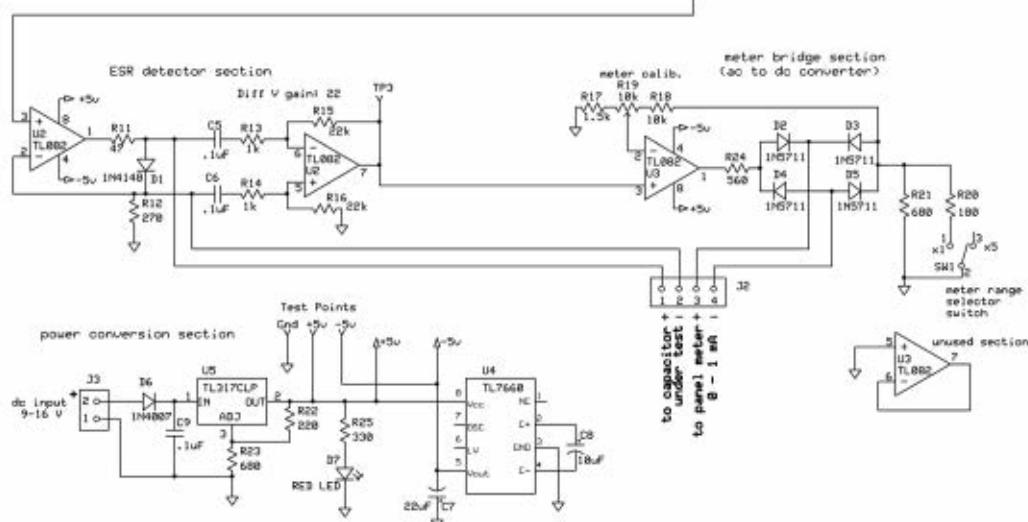


FIGURE 2. Electrical schematic of the ESR meter.



Since the phase-shift oscillator circuit has a moderately high output impedance, this prevents loading of the oscillator circuit. There is also a gain-control potentiometer (R8) which allows you to adjust the level of the 100 kHz signal. Resistors R6 and R7 insert a small DC offset on the AC from the oscillator, so that the signal passed on to the ESR detector has a slight positive bias. Since this signal is applied to the capacitor being tested, some DC bias is required for polarized capacitors.

The circuit path between the oscillator and the buffer amplifier passes through switching front panel 3.5 mm mono jack J1. The jack is wired so that an external AC source plugged in here will interrupt the built-in 100 kHz oscillator and act as a substitute for it. This feature allows you to measure ESR at different frequencies, if you so desire.

If you're interested in a detailed explanation of how the phase-shift oscillator works, you can find a pdf at the article link.

ITEM	Description	Mfr/Part Number
C1, C2, C3	1 nF, 100V, ceramic	Vishay K102K10X7RH5UH5
C4, C5, C6, C9	0.1 μ F, 50V, ceramic	Vishay K104K10X7RF5UH5
C7	22 μ F, 16V, tantalum	Kemet T350F226K016AT7301
C8	10 μ F, 35V, tantalum	Kemet T350G106K035AT7301
D1	1N4148	
D2, D3, D4, D5	1N5711 Schottky diode	
D6	1N4007	
D7	Red LED	
J1	3.5 mm switched jack	
J2	Four-pin male header	CUI MJ-3502N
J3	Two-pin male header	
R1, R15, R16	22K	
R2	1 meg	
R3, R4, R5	2.2K	
R6	100K	
R7	820K	
R8, R19	10K trimmer	Bourns 3339P-1-103LF
R9, R18	10K	
R10	0	[wire jumper]
R11	47	
R12	270	
R22	220	
R13, R14	1K	
R17	1.5K	
R20	180	
R21, R23	680	
R24	560	
R25	330	
SW1	SPDT toggle switch	C&K 7101SD9ABE
TP1, TP2, TP3	Test point	[none]
U1, U2, U3	Dual op-amp	Texas Inst. TL082CP
U4	Voltage inverter	Texas Inst. TL7660CP
U5	Adj. voltage regulator	Texas Inst. TL317CLP
(4) Eight-pin DIP IC sockets (optional)		
Panel meter 0-1 mA		
(2) binding posts		
Rocker switch (power switch), SPST		
DC coax jack 2.1 mm		
Printed circuit board 2.5 x 3.8"		
Enclosure 3 x 4 x 5"		
NOTE: All resistors are axial lead, 1/8 watt or higher.		
CUI PJ-011A		
ExpressPCB		
Hammond Mfg. 1411-LU		

The ESR Detector

This is it, folks! This is where most of the action takes place. The first section of op-amp U2 is a voltage-to-current converter where the 100 kHz signal from the oscillator is converted to a current of about 7 mA peak-to-peak. The Capacitor Under Test (CUT) is connected inside the feedback loop of this stage via two front panel binding posts, so the same current flows through the CUT.

Diode D1 — in parallel with the CUT — provides a discharge path for the CUT when you connect it to the ESR meter in case it's already charged up. In normal operation, the voltage across the CUT is so low that D1 never turns on, so has no effect on the operation of the circuit.

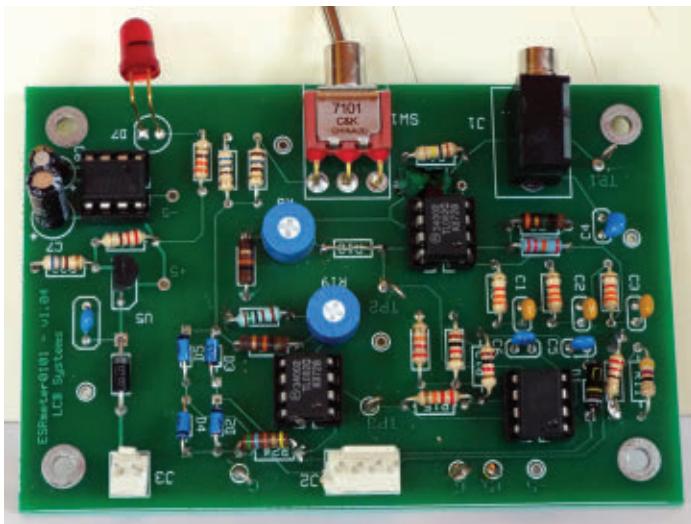
Now that we have established a known AC current through the CUT, it only remains to measure the voltage developed across it. The magnitude of this voltage is directly proportional to the ESR of the CUT. The ESR is usually very low — a few tens of ohms at the most — so this voltage will be down in the millivolt range. The second section of U2 is configured as an AC coupled differential amplifier with a gain of 22, which raises the AC component of the voltage across the CUT to a more convenient level for the meter amplifier stage.

The Meter Amplifier

I wanted the ESR to be displayed on a conventional 0-1 mA moving coil panel meter. (It's my own personal taste.) For an instrument like this, I just prefer the look of a traditional panel meter over a numerical digital readout. For this to happen, the AC voltage from the ESR detector must be appropriately scaled and converted to a DC current. This is the job of U3 and the diode bridge D2-D5.

The AC from the ESR detector — which represents the level of the ESR we are trying to measure — is fed to op-amp U3. The output of U3 passes through R24, through a

Parts List



■ **FIGURE 3.** Printed circuit board.

bridge circuit composed of Schottky diodes D2-D5, and through current-sensing resistors R20 and R21 to ground. The voltage developed across these resistors is fed back to the inverting input of U3, thus completing the feedback loop.

Within the diode bridge, the AC is rectified and passed through the front panel meter, which responds only to the average (i.e., DC) component. By enclosing the bridge within the op-amp feedback loop, most of the nonlinearities inherent when a bridge is used to drive a moving coil meter are removed.

Switch SW1 puts R20 in parallel with R21, reducing the value of the current-sense resistor combination, thus increasing the sensitivity of the meter. With SW1 closed, the full scale sensitivity of the ESR meter is one ohm. With it open, an ESR of five ohms is required to drive the meter to full scale.

The gain of this stage is set by R17, R18, and R19. The latter is a 10K ohm trimmer potentiometer used to set the calibration of the ESR meter after the circuit is built.

If the ESR instrument is powered up with no CUT connected, R24 limits the average current through the panel meter to a maximum value of about 2 mA, thereby making life a bit easier for the meter.

The Power Conversion Section

In this design, I chose to provide both +5V and -5V power buses for the op-amps. This simplifies the circuit design and makes it easier to follow, in my opinion. A single-supply approach would require the additional complication of providing a virtual ground reference throughout the ESR meter. A conventional three-terminal voltage regulator at the input U5 supplies the +5V bus. The -5V bus is easily supplied by U4 – a dandy



■ **FIGURE 4.** ESR meter after calibration. The meter is displaying the value of the one ohm test resistor.

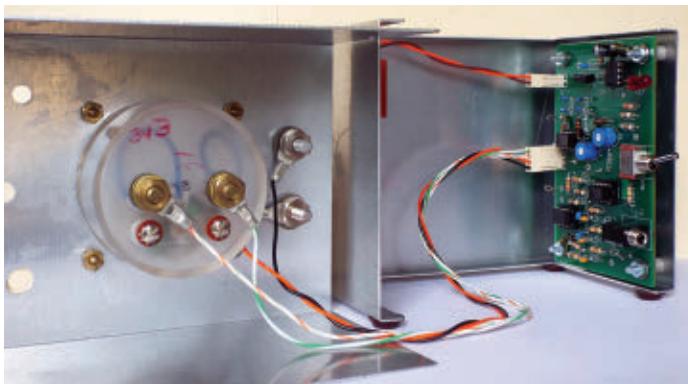
component from Texas Instruments (TI) that conveniently puts out a DC voltage equal in magnitude to its input, but with a reversed polarity.

Construction

I used the services of ExpressPCB (www.expresspcb.com) to lay out and fabricate the printed circuit board (PCB) for this project. Their standard low cost MiniBoard fits very nicely into a 3 x 4 x 5 inch aluminum enclosure, with plenty of room for a 0-1 mA meter and two binding posts to be mounted on the front panel. The PCB (shown in Figure 3) is laid out with J1 (the external source connector), SW1 (the meter range switch), and D7 (the power-on LED) along one edge. The PCB is



■ **FIGURE 5.** The ESR meter in action, reading the ESR of an old (date code 1966) 100 μ F tantalum capacitor as 0.3 ohms.



■ FIGURE 6. Internal wiring, showing the mounting of the circuit board and the cabling to the front and rear panels.

What a Capacitor Really Looks Like

Nothing is perfect in this world, and that includes electronic components. Resistors have a little bit of capacitance and inductance; inductors have a smidgeon of resistance; and capacitors have all of the above. Fortunately, most of the time these "parasitic" quantities can be ignored and we can treat the components we use as ideal resistors, inductors, and capacitors.

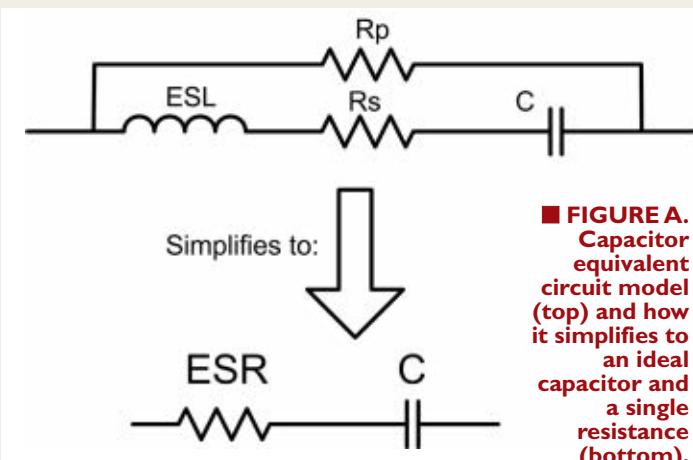
Notice I said "most of the time." Capacitors — especially large value electrolytics — can suffer from an illusory low value resistor that appears to be in series with an ideal capacitor. This is referred to as the Equivalent Series Resistance (ESR) of the capacitor. It's "illusory" because ESR is not a true resistance; rather, it's the result of a combination of many factors — all of which contribute in some way to power loss in the capacitor. **Figure A** is the equivalent circuit model of a typical real world capacitor and gives a better picture of what I'm talking about. For high value capacitors and at low frequencies, the stray inductance shown in the model can usually be ignored and the two resistances combined into one.

Since you're reading this magazine, you probably already know that every capacitor is basically just a pair of conductors separated by a dielectric. The conductors in a large value electrolytic capacitor are usually strips of foil. The dielectric is an insulating oxide layer formed on one of the strips (the "anode," or positive electrode), plus a liquid or paste electrolyte which acts as the second electrode of the capacitor (the "cathode"). This stuff can be corrosive, so if you have a capacitor which is physically damaged and oozing electrolyte, be careful of getting it on your skin.

Losses in the dielectric plus leakage across the capacitor and resistance in the welds and mechanical crimp contacts to the terminals all contribute to the ESR.

Here's the problem: Over time — especially at elevated temperatures — the liquid electrolyte component of the dielectric dries (or leaks) out. The capacitance may not change very much, but there will be an increase in resistivity; therefore, the ESR rises. To make matters worse, depending on the dielectric substance, the ESR can vary with frequency. This can be a problem if the capacitor must handle substantial alternating current, as in a switching power supply, for example. High ESR combined with high current means extra power dissipated in the capacitor. The resulting temperature rise can cause further degradation and premature failure.

Aluminum electrolytic capacitors are particularly prone to this problem — especially if they've been around for a long time. Solid tantalum capacitors also have ESR problems but to a lesser degree. Small ceramic capacitors are essentially free of this plague.



■ FIGURE A. Capacitor equivalent circuit model (top) and how it simplifies to an ideal capacitor and a single resistance (bottom).

mounted on 1/4" standoffs on one wall of the enclosure, with appropriate holes bored in the front panel to allow access to these three components. Refer to **Figures 4, 5, and 6**.

The ExpressPCB schematic and PCB files can be found at the article link.

Each of the test points for ground — +5V, -5V, TP1, TP2, and TP3 — is made of a short length of solid hook-up wire. One end is soldered into a hole in the PCB, and the free end is formed into a loop for easy grabbing by clip leads or test probes.

Figure 6 is an inside view of the enclosure, showing the internal wiring. Here you can see that connections to the front panel meter and binding posts are brought out from the PCB by four-pin male connector J2, and power from the rear panel via two-pin male connector J3.

Raw DC power (9 to 16 VDC) is supplied through a 2.1 mm coax jack and SPST rocker switch on the rear panel as shown in **Figure 7**. The current requirement is fairly modest. The whole circuit runs on less than 40 mA. A good quality wall wart type of power source works very well, as does a 9V alkaline battery.

The front panel label sheet and a new face for the panel meter were drawn using Microsoft Visio, printed on heavy paper stock, and glued in place.

Setup and Calibration

There are two adjustment trimmer potentiometers on the circuit board. One (R8) is used to adjust the output of the phase-shift oscillator to about 1.8V peak-to-peak, and the other (R19) sets the meter sensitivity. Full details of this procedure can be found in the downloads at the article link.

Figure 4 shows the result of this setup with a one ohm resistor connected across the CUT binding posts. In **Figure 5**, a 100 μ F tantalum capacitor is being measured for ESR.

Final Notes

Most projects hit a snag or two along the way, and so did this one. If you look carefully, you may spot a small inconsistency between the photo of the printed circuit board in **Figure 3** and the ExpressPCB layout file included in the online files. This is the result of an initial design goof on my part, which required me to cut a couple of PCB traces and re-locate components R7 and C4. I revised the PCB layout after the fact, and the ExpressPCB layout file at the article link has these corrections and agrees with the schematic.

This meter is – in principle – suitable for checking a capacitor's ESR without removing it from the equipment it's connected to. The impedance of the surrounding circuitry is normally much higher than the ESR being measured, and the voltage developed across the CUT is quite small: less than 100 millivolts – much too low to switch on any semiconductor junctions in the vicinity. Power to the equipment should be off, of course, and the ESR meter should probably be running off an isolated power source like a 9V battery. I have not tried this type of measurement myself, but I see no reason why it would not be successful.

At this point, I would like to mention some limitations of this instrument, or of almost any ESR meter:

- 1) This meter is not suitable for testing capacitors less than 30 microfarads. If the CUT is too low, the reactance at the measuring frequency becomes significant, resulting in an excessive ESR reading. The solution to this problem is to redesign the system to use a higher frequency. If the need arises, I may try this as a future project.

- 2) A capacitor with an internal short circuit will appear to have a misleadingly low value of ESR, so don't be fooled (as I have been). Check with a DC ohmmeter if there's any doubt.

- 3) Because an ESR meter is essentially a low range ohmmeter, long test leads from the CUT can contribute errors to the ESR reading.

- 4) ESR can depend on external factors such as temperature or applied voltage, so a capacitor may behave a bit differently in a real circuit than when it is being tested all by itself.

- 5) Although this unit has some protection built into it, applying a fully charged high value capacitor to the test terminals could damage the circuitry. It's always a good idea to



■ FIGURE 7. Rear panel of the ESR meter.

manually discharge a capacitor before testing.

One final remark: ESR measurement does not usually require a high degree of accuracy, and the meter described in this article should be adequate for routine troubleshooting. In my case, it was very helpful in identifying questionable components, possibly saving me some hair-pulling/teeth-gnashing frustration on a future project. **NV**

An advertisement for Probotix CNC Routers & Control Systems. The top half of the ad features the Probotix logo in blue and purple, followed by the text "CNC Routers & Control Systems". Below the text is a photograph of a CNC router machine with a green tool mounted on its Z-axis, positioned over a light-colored work surface. To the right of the machine is a QR code. The bottom half of the ad has a dark background with the text "When You Are Serious About CNC" in blue, followed by the website "www.probotix.com" in white.

Rock It with a MIDI KEYTAR



The Rock Band 3 keyboard (keytar) can currently be found online for little more than \$20. Although primarily designed as a game controller for the Rock Band video game, it also doubles as a MIDI controller. This makes it one of the least expensive options available for a MIDI keyboard to musicians. At first glance, the MIDI options seem rudimentary at best, and I ignored buying one when they first came out. However, there are quite a few options available that aren't readily apparent, and even a couple hidden controller features that I've chosen to exploit. The end result is a very inexpensive — yet very capable — MIDI controller that can be used both in the music studio and for a little fun on a live stage.

Rock Band Keyboard MIDI Features Summary

The stock Rock Band keyboard is shown in **Figure 1**. Let's take a look at the features available right out of the box:

- Velocity-sensitive two octave keyboard with octave selection up and down.
- Modulation (vibrato) ribbon controller with a button-selectable pitch bend function.
- MIDI program up and down selection.
- MIDI sequencer controls: Stop, Continue, and Start to run your DAW (Digital Audio Workstation), drum machine, or sequencer.
- MIDI panic command (all notes off) for those annoying stuck notes.
- MIDI Drum mode where you can play drum sounds on an external module on MIDI channel 10 (normally, it only outputs notes on MIDI channel 1).

In addition, there is a small 1/8" (3.2 mm) TRS (tip, ring, and sleeve; sometimes called a 'stereo jack') jack next to the MIDI out connector that was intended for future controller expansion. The planned accessory was a dual foot controller that would add a sustain pedal and a volume/foot controller potentiometer pedal. The function of the volume/foot controller is selected from the 'D-pad' gaming control on the front panel. The D-pad is the four-position flat joystick control on the front panel which offers switch closures for up, down, left, and right movements.

In summary, **Table 1** is a list of the front panel MIDI functions (note some buttons are labeled differently between the Xbox, Sony PlayStation, and Wii versions).

You may want to label these MIDI functions like I did as shown in **Figure 2**. Since I painted the body of mine, relabeling was needed anyway. If you do plan to paint the body on yours, remove everything from the case. That will make painting much easier, and prevent potential issues with the electronics.

Let's Add a MIDI Sustain Switch

Now comes the fun part: Tapping those features in the 1/8" pedal jack. I think most will agree that a sustain pedal on a two octave keyboard is not very useful. However, when used on a stage as a keytar, a neck-mounted sustain switch is quite useful for theatrics, and a look at most higher-end keytars reveals this is a fairly common feature. We *could* add another switch to the neck for this purpose, but there is already a nice momentary switch mounted on the neck in an ideal position. The function of the current switch is not very useful, however. When held down, it causes the ribbon controller — usually controlling modulation — to instead control pitch bending. It's quite awkward to do pitch bending this way, so I came up with another method.

I added a small SPST toggle switch to an unused area on the front of the body. I then desoldered the wire from the current neck switch (labeled 'overdrive' on the neck printed circuit board [PCB]) and soldered it to the toggle switch. I then ran ground to the other terminal of the

- D-pad up = (Drum mode) lowest keyboard octave outputs on MIDI channel 10 for playing drum voices.
- D-pad down = pedal outputs MIDI channel volume cc 7.
- D-pad right = pedal outputs foot controller MIDI cc 4.
- D-pad left = pedal outputs expression MIDI cc 11.



■ **FIGURE 1.** Stock Rock Band 3 keyboard.



■ **FIGURE 2.** Repainted keyboard with MIDI functions labeled.



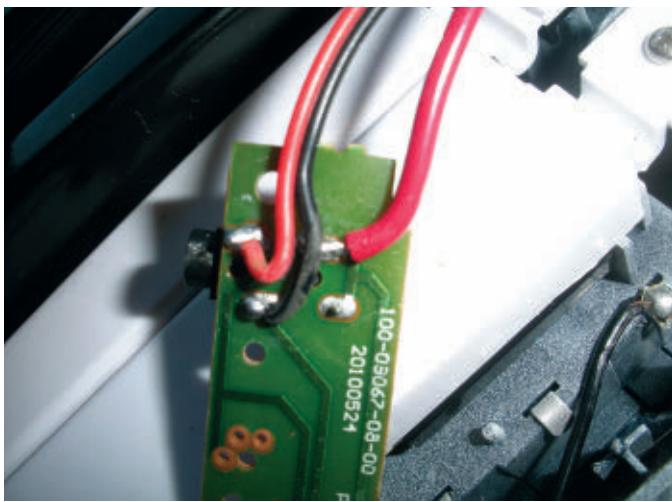
■ **FIGURE 3.** Neck PCB. To start the MIDI sustain switch function and move the pitch bend/modulation selection switch, desolder and remove the top wire in the ribbon cable. It's labeled "over_d" on the PCB. Next, solder a wire on the underside of the "gnd" wire solder connection next to the "overdrive" hole.

switch. The ground is conveniently located next to where the overdrive wire was on the neck PCB. Details are shown in **Figure 3**. When the switch is closed, pitch bend mode is activated on the ribbon controller. An open switch defaults to modulation mode.

To convert the existing neck switch into a momentary sustain switch, I ran a wire from the 1/8" TRS pedal jack's

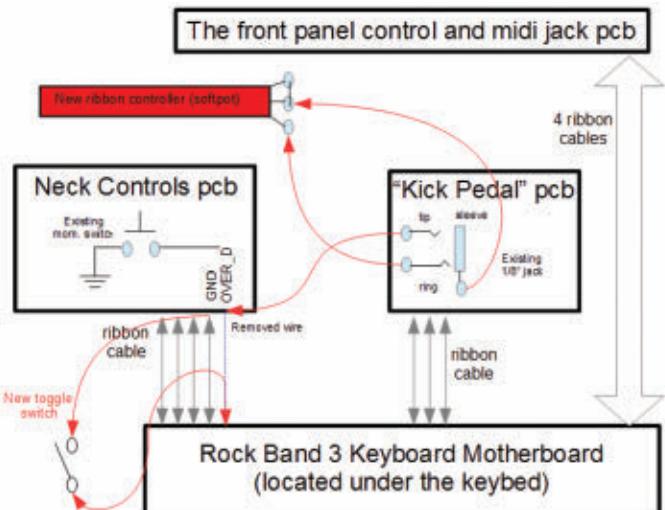
Table 1.

MIDI Command	PlayStation	Wii	Xbox
MIDI Sequencer Stop	Select	- (minus)	Back
MIDI Sequencer Continue	PS Logo	Keyboard	Xbox logo
MIDI Sequencer Start	Start	+ (plus)	Start
MIDI Panic (all notes off/reset)	All 3 buttons above	All 3 buttons above	All 3 buttons above
Keyboard octave decrease by one	(square)	1	X
MIDI Program increase by one	(triangle)	2	Y
Keyboard octave increase by one	(circle)	B	B
MIDI Program decrease by one	X	A	A



■ **FIGURE 4. "Kick Pedal" PCB.** Run three wires from the tip, ring, and sleeve terminals to the neck area of the keyboard. The ring is the ground connection and will need to route to the minimum value side of the softpot. The left red wire is the sleeve terminal and connects to the softpot maximum value side of the softpot. The middle black wire is the ring terminal, and the right red wire is the tip terminal which routes to the neck PCB for the new sustain switch.

tip terminal to the now-vacant overdrive neck PCB hole. The 1/8" TRS jack is mounted on a small board to the right of the highest key and is labeled "kick PCB." I ran three wires from this board up to the neck area: tip, ring, and sleeve (Figure 4). The ring and sleeve wires are used



■ **FIGURE 5. Overall block diagram of the Rock Band 3 keyboard with modifications.**

for the next part of our modification.

The 1/8" TRS jack is wired a little differently than you might expect. The tip is the sustain signal input. It is held high and looking for a digital 0 (ground) to signal the keyboard's electronics. The sleeve is the AD input for the volume/foot controller potentiometer pedal. The ring is the ground.

The tip is looking for an open circuit to ground connection (switch closure) to send a sustain pedal on command. A ground to open circuit (switch release) signals a sustain pedal off command. The sleeve AD input is looking for a resistance in the 100 to 10K ohm range. The keyboard's software also complicates things.

If you go to a resistance below about 100 ohms, the software tells the keyboard there is no potentiometer pedal and the volume defaults to full. In this scenario, you could be decreasing the volume and then when you are just about at zero volume, the thing will call for full volume. I suppose this could be used for dynamic effect, but I just found it annoying.

Also, if the resistance is too high, the software tells the keyboard nothing's plugged in and the functions are disabled. Figure 5 shows the overall electronics block diagram with the modifications.

Let's Add Another MIDI Controller

The ideal potentiometer to replace the volume pedal for me is a 50 mm long (2") 10K ohm softpot. Softpots are made by Spectra Symbol and are available at various electronic distributors such as SparkFun, Digi-Key, Adafruit, and Mouser Electronics. Mine cost about \$5. You can get longer lengths, but 50 mm seemed the right length for the

Resources

Rock Band 3 MIDI Guide:

[www.fakebitpolytechnic.com/
wp-content/uploads/2013/07/rb3keybordmidimanualv2.pdf](http://www.fakebitpolytechnic.com/wp-content/uploads/2013/07/rb3keybordmidimanualv2.pdf)

Softpot Datasheet:

www.sparkfun.com/datasheets/Sensors/Flex/SoftPot-Datasheet.pdf

Sources for Softpot:

www.sparkfun.com
www.adafruit.com
www.mouser.com
www.digikey.com

Sources for further information on the Rock Band 3 keyboard, other modifications, adding sound capabilities, using with softsynths, etc.: <http://createdigitalmusic.com/2010/10/hands-on-rock-band-3s-keytar-a-surprisingly-serious-80-midi-keyboard>

<http://samislam.hubpages.com/hub/Rockband-3-Keyboard-as-a-MIDI-Controller>

<http://hackaday.com/2011/02/18/8-bit-midi-synthesizer>
<http://robertsonics.com/2014/10/19/turn-the-rockband-3-keyboard-into-a-sampling-instrument-with-the-wav-trigger>
www.youtube.com/watch?v=HGoSdSSI4ul

Sites for other DIY Electronic Music Ideas:

www.muffwiggler.com/forum
<http://electro-music.com/forum/index.php?f=112>

RB3 Keyboard Controls:

Velocity note data
MIDI program up
MIDI program down
Octave up
Octave down
MIDI panic (three finger salute)
Mod wheel (pitch bend wheel selectable)
MIDI sustain (jack input)
MIDI foot controller, volume control, expression control (jack)

input/selectable)
MIDI start, stop, continue sequencer control

Mods:

1. Add a toggle switch to select ribbon mod or pitch bend.
2. Convert flat neck switch into a sustain switch.
3. Add a pot on the neck for foot controller/ expression/ volume.

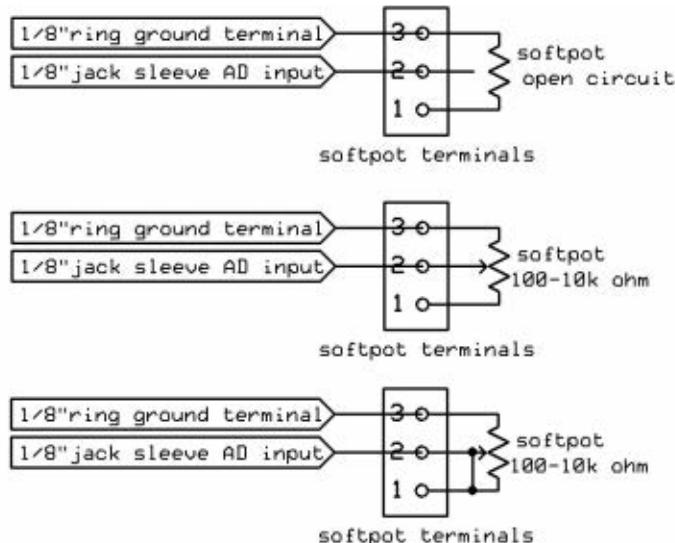


FIGURE 8. Softpot operation. The top picture shows how the circuit appears without a finger pressing on it: The wiper is open circuit. The middle picture shows that the softpot behaves like a normal potentiometer with a finger applying pressure to it. The bottom picture shows how we short the wiper to the maximum value softpot terminal to ensure the keyboard's AD input always sees at least 10K ohm of resistance — even when the finger is removed.

keytar neck. I had to trim about 1/8" (3.2 mm) off each side of the softpot to fit on the top half of the body.

Be careful not to cut into the traces of the pot if you do elect to trim it. The back of the softpot comes with an adhesive to mount it. I routed the connector ribbon into the body by cutting a thin slot perpendicular to the upper case half with a hack saw (**Figure 6**).

The width of the hacksaw blade was more than adequate for the connector to slip into.

To wire up the softpot (**Figure 7**), I connected it to the two remaining wires from the TRS pedal jack. If you're careful, you can snake the wires underneath the keyboard where they are protected and out of the way. One wire connects to the ring (ground) and the other wire hooks up to the sleeve (AD input). Be sure to run the ground to the



FIGURE 6. Softpot mounted on the keyboard neck. Note the hacksaw cut made to accommodate the softpot's flat ribbon cable terminals.

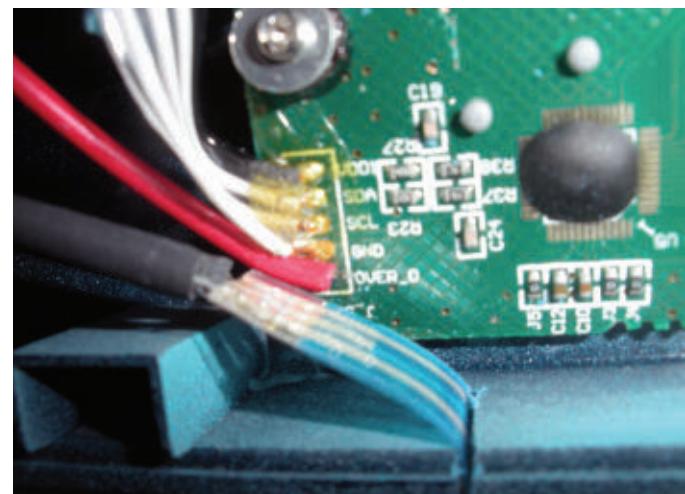


FIGURE 7. Softpot wiring detail. Note shrink tubing applied to the terminals to prevent short circuits. Also note the new red sustain wire connection to the neck PCB at the 'over_d' location.

minimum side of the softpot. Don't forget to protect any exposed terminals with shrink tubing. Also take care to short the minimum softpot terminal to the middle/wiper terminal so it operates correctly.

Don't just hook the AD input to the middle wiper and the maximum value. The wiper goes open circuit when a finger is not applied to it. Remember the goofy behavior I mentioned with the software? By shorting the minimum terminal and wiper terminals together, you ensure the AD input always has at least 10K ohms of resistance and doesn't see an open circuit.

I should mention the softpot has a minimum resistance of 100 ohms, which takes care of the AD input not liking to see less than 100 ohms on its input.

Figure 8 shows the normal softpot operation and how

it should be wired in the Rock Band keyboard. If you elect to use a traditional potentiometer that goes close to zero ohms at the minimum setting, you'll want to wire a 100-150 ohm resistor between the minimum softpot terminal and the TRS jack's ring/ground connection.

The function of the new softpot is selected from the 'D-Pad.' It will output MIDI expression, foot controller, or MIDI volume data (MIDI controllers 11, 4, and 7, respectively).

I use my software DAW/sequencer/tracker's 'learn' features to match the softpot to the feature I want to

control, so the setting in the keyboard is not critical.

Watch Your Grounds!

During the building and testing of this project, I noticed not all grounds in the Rock Band 3 keyboard are created equal. There is a digital ground, an analog ground, and each board has its own ground. The ground wiring I'm specifying for these modifications seems to be the most robust I've been able to come up with. Dying batteries even created a situation where I had to switch grounds to keep things operating intelligently longer.

Test It Out

After re-assembling the keyboard's body, put the batteries in, plug in a MIDI cord, and attach it to your favorite synthesizer module or MIDI-to-USB converter for computer use. Then, turn it on and you're ready to make music.

The original ribbon controller can be utilized by your left index finger; the new sustain button with your middle finger; and the new added softpot with your thumb. Now, you have the expression control of the more professional keytar for about \$30.

At that price point, every keyboardist should have one in their arsenal! **NV**

PCBShopper

PCBShopper.com is a price comparison site for printed circuit boards. Enter your board's specs and PCBShopper will show you prices and delivery times from 25 manufacturers, including many of the best-known companies in Asia, North America and Europe.

**Electronix
EXPRESS**

Your Single Source of Electronic Equipment, Tools, and Components

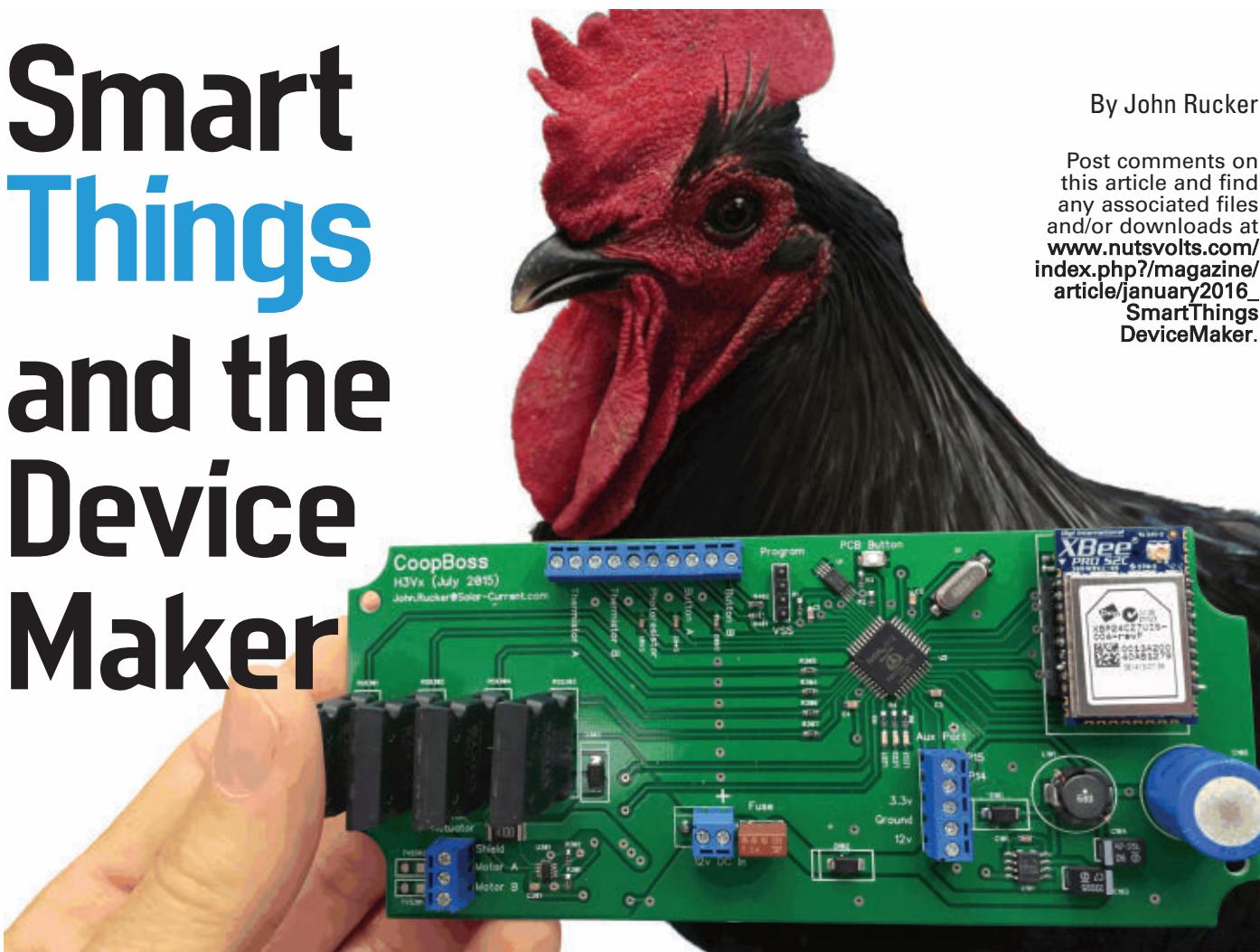
<p>01DM820B Digital Multimeter Super Economy</p> <p>DMM - Volts, current, Resistance, Transistor and Diode Test. Includes Leads</p> <p>\$7.50</p>	<p>01DMMY64 Digital Multimeter 32 Ranges</p> <p>Including temperature, capacitance, frequency and diode/transistor testing</p> <p>\$29.95</p>	<p>01DMM9803R Bench DMM True RMS</p> <p>Auto and manual range, RS232C standard interface. Runs on DC or AC</p> <p>\$139.95</p>	<p>Sweep Function Generator 3MHz, 6 Waveform Functions, Int/Ext Counter, lin/log sweep</p> <p>Model FG-30 \$135 (no display)</p> <p>Model FG-32 \$195 (5 digit display)</p>
<p>01DS1102E 100MHz Rigol Oscilloscope</p> <p>Features: 1 million points of deep memory, FFT, record and replay, roll mode, alternate trigger mode, and adjustable trigger sensitivity</p> <p>\$399</p>	<p>01SDS1052DL Siglent 50MHz DSO</p> <p>500MSa/s Sampling 32K Memory 6 Digit Frequency Counter USB Interface 7" Color TFT-LCD Screen</p> <p>\$279</p>	<p>DC Power Supplies</p> <p>Model HY3003 Variable Output, 0-30VDC, 0-3Amp.</p> <p>Model HY3003-3 Two 0-30VDC, 0-3Amp Variable Outputs plus 5V 3A fixed.</p> <p>\$95</p>	<p>0603ZD98 48W Soldering Station Temp. Controlled 310°F~840°F</p> <p>\$15.25</p>
<p>Alligator Leads</p> <p>Set of 10 \$3.55</p>	<p>Switches</p> <p>8 POS DIP (V17DIP8SS) Toggle Mini SPDT (17TOGSD-M) Toggle Mini DPDT (17TOGDD-M)</p> <p>\$0.90 \$0.85 1.40 1.20 1.55 1.35</p>	<p>Potentiometers</p> <p>1-9 10+ Cermet (STS Series) Multiturn (MTT Series) Panel Mount (PMA Series)</p> <p>\$0.85 \$0.75 0.70 0.60 0.45 0.80 0.70</p>	

1 (800) 972-2225 | <http://www.elexp.com> | contact@elexp.com

Smart Things and the Device Maker

By John Rucker

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?/magazine/article/january2016_SmartThings_DeviceMaker.



I'm a device maker at heart. What I mean is I love to create dedicated microcontroller solutions that communicate with the real world. Some of my devices have turned into commercial products, but most don't and my wife ends up enjoying them around the house. Well, I'm not so sure "enjoy" is the correct term ... maybe tolerates would be more appropriate.

About two years ago, I developed a C# application for work that allows a Windows PC to talk directly to a Hue light bulb over the ZigBee HA protocol. My app changed the bulb's color based on a user's presence information in Microsoft Lync (Skype for business). If the user was busy, the Hue bulb would be red; if they were available, it would be green, etc.

This project opened my eyes to what is possible when a device is connected to the Internet of Things (IoT). I created this application without any documentation on the Hue bulb. I simply queried the bulb for its capabilities. When asked, the bulb responds back with "I'm a light and I can be turned on or off, my brightness can be changed, and you can change my color." The device tells me what it

can do and how to talk to it!! This is all built into the ZigBee Home Automation public profile.

I couldn't believe how well designed this was, and read everything I could get my hands on about ZigBee and home automation. I purchased and played around with a few home automation hubs trying to get a feel for how I could use them to control my devices, only to be disappointed on how closed they were. Then, it happened. A dream come true.

I stumbled across a young startup: SmartThings. A home automation platform built for developers from the ground up. They have a cloud-based development environment that allows you to create smart apps and custom device types to talk to, and ... wait for it ... yes,

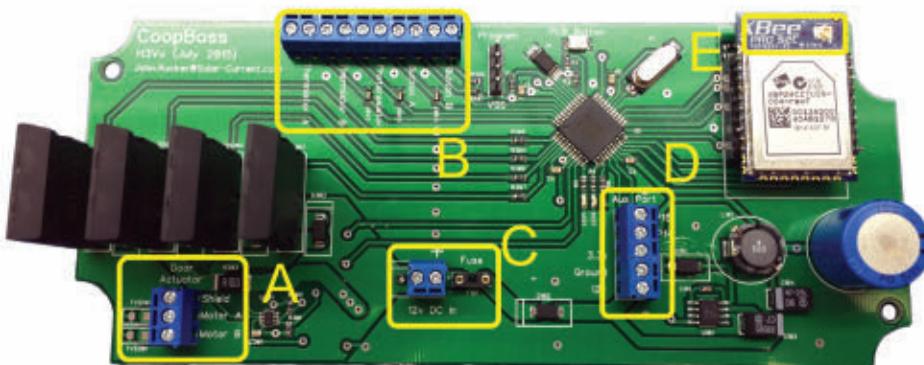


FIGURE 1. CoopBoss Parts.

microcontroller-based devices over ZigBee!! They even have an Arduino shield you can use to hack together a prototype.

It was the perfect storm. I had an open platform (SmartThings), an open protocol to talk to it (ZigBee), and years' worth of awesome Propeller-based devices that could take advantage of it. So, last April, I created a standards-based ZigBee object for the Parallax Propeller that allows my devices to communicate based on the ZigBee HA public profile. It opened a huge door of opportunity for my devices. Now, I can make a device, connect it to the SmartThings hub over ZigBee, and control it with my iPhone. All this and I don't have to write one line of code that runs on the iPhone!! My device identifies itself to the SmartThings hub, tells it what it is capable of, and the SmartThings hub then determines how to control it.

Recently, one of my ZigBee HA projects – the CoopBoss (Chicken Coop Door controller) – won Best in Show at SmartThings and it's on its way to becoming a commercial product. They did a nice write-up on the CoopBoss; if you're interested, you can check it out at <http://blog.smartthings.com/stories/a-smart-chicken-coop>.

The CoopBoss was a project I put together after my

wife lost three of her best hens in a mink attack one night last spring. We forgot to close the coop door and a mink got inside and just ripped three of her five chickens apart. She was heartbroken. Her chickens had been with us for three years supplying my lunch three times a week.

The CoopBoss is a custom printed circuit board (PCB) that drives an off-the-shelf 12V actuator to push the coop's sliding door closed at sunset and open at sunrise. My wife is able to control

it with her smartphone, and can open or close the door anytime, anywhere she has service.

Figure 1 shows the ports to connect the CoopBoss to the real world. They are the following:

Motor Connection: Connects to a 12V DC linear actuator to push and pull the coop's door. The motor control circuit has been tested with Progressive Automations' 12V actuator model number PA-14-12-35. However, just about any 12V DC four amp or less actuator should work.

Main I/O Header: This header is connected to two thermistors, one photoresistor, and two normally-open pushbuttons.

DC In: 12V DC power in and five amp fuse holder.

Aux Port: Auxiliary port for future expansion.

Antenna: XBee antenna connection.

Figure 2 highlights the CoopBoss' onboard modules which include:

Motor Control: Based on IXYS Integrated Circuit's CPC1709J solid-state relays. The motor control circuit uses four optically coupled solid-state relays arranged in an H-bridge to drive the 12V DC actuator motor forward and reverse. It is connected to the microcontroller (H) with four I/O pins.

Object Detection: Motor current monitoring is based on the Texas Instruments' INA219BIDCNT bi-directional current/power monitor chip. If the door bumps up against an object during a close, this circuit will detect the small increase in current and quickly relay that to the microprocessor (H) over an I²C bus. The

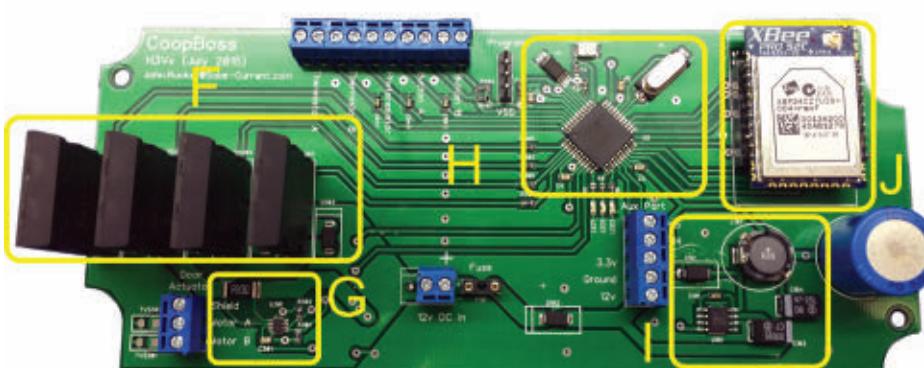


FIGURE 2. CoopBoss Modules.

A ZigBee device supports several clusters or the same cluster several times each on a unique ZigBee endpoint. If you have a device with multiple LEDs to control, each LED's On/Off Cluster can be on a unique ZigBee endpoint. ZigBee treats the cluster number and endpoint as part of the device's destination address. This way, you cannot only direct a packet to a device, but to the correct application within the device. We will see examples of ZigBee addressing in a follow-up article when we build out our circuit that controls an LED through a ZigBee Home Automation network.

microprocessor will then instruct the motor control circuit (F) to stop closing the door and open back up. This all happens in just a few milliseconds.

Microcontroller: The microcontroller is based on the Parallax P8X32A-Q44 multicore Propeller. The Propeller's multiple core processors are dedicated to monitoring and controlling the various modules of the CoopBoss. For example, to get almost near instant object detection, one processor is dedicated to monitoring the object detection (G) circuit, while the other processors are free to process ZigBee traffic, button I/O, read coop temperature, and monitor the ambient light level.

PCB Power Supply: The power circuit is based on the Texas Instruments' LM2675MX-3.3 switching regulator. It has a very wide input range from 8V to 40V DC and outputs 3.3V DC with up to one amp of current. The power circuit also has a one farad supercapacitor to help regulate the power through high current demands from the motor control circuit (F).

ZigBee Radio: The ZigBee radio is from Digi International; the part number is XBP24CZ7UIS-004. This radio securely communicates with the SmartThings hub that connects to the SmartThings cloud on the Internet. A smartphone app (SmartThings Custom Device Type) allows secure control of the CoopBoss over the Internet.

The CoopBoss is a good example of how a reliable circuit can be enhanced by connecting it to the IoT. By adding a ZigBee radio and support for the ZigBee Home Automation profile, the CoopBoss now can have an intuitive user interface accessible from a smartphone anywhere the smartphone has Internet access (Figure 3). By using standard ZigBee clusters with custom attributes, you can do some very slick stuff like add sliders that allow the user to pick the sunlight level for a close or an open. You can even set the sensitivity of the door's object detection circuit, so if a chicken is in the way, the door will bump up against her and open back up.

This project and several others like it have been very rewarding. I would like to share my experiences with focus on the ZigBee communications. The real story here is not the circuit, but how you can take a simple circuit and make it into a full blown solution by connecting it to the IoT. To accomplish this, I have laid out a simple "hello world" test circuit in "The Controlling a Custom Device

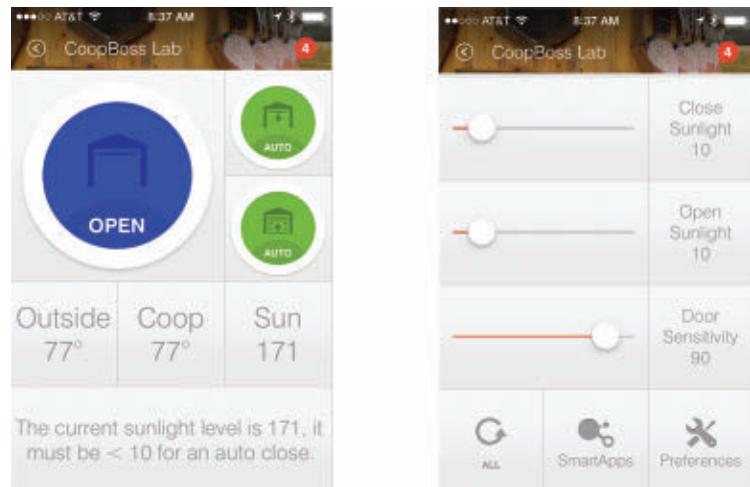


FIGURE 3. Smartphone App.

with SmartThings" section of this article. It's an LED tied to a Propeller and a ZigBee radio. We will gradually add complexity to the circuit as we go deeper into the protocol.

Why ZigBee?

I would like to take a step back and acknowledge that ZigBee is not the only show in town. Frankly, there is a lot to choose from when it comes down to picking a protocol for IoT communications, but arguably, the big players today are Wi-Fi, Bluetooth LTE, Z-Wave, and ZigBee HA. Almost all the popular home automation hubs support one or all of these protocols. It's beyond the scope of this article to go into technical detail on each protocol, but I will expand on why I selected ZigBee as the communication standard for my devices.

Self-Configuring, Self-Healing Mesh Network

ZigBee is a mature, open solution based on a self-configuring mesh network. That is important when your device is outside several hundred feet away from your ZigBee hub. If it is too far away to get a good signal, you

The CoopBoss' PCB is designed to fit inside a weather-tight aluminum enclosure with weather-tight grommets.



Table 1.

Cluster Number	Description
0x0006	On/Off Cluster for devices that can be turned on, off, or toggled.
0x0008	Level Cluster for a device that can be dimmed.
0x0101	Door Lock Cluster for controlling a door and its lock.
0x0402	Temperature Measurement Cluster.

represented as hex numbers. **Table 1** is a very small sample of common clusters used by devices.

Cluster commands are either a server or client command. For example, say we have a ZigBee enabled light bulb and a ZigBee enabled wall switch. Both devices will communicate using the commands defined in the On/Off (0x0006) cluster. The light bulb will be the server and the light switch will be the client. How the light switch sends the “on” command and how the bulb receives the “on” command are all defined in the On/Off (0x0006) cluster specification.

Why is this important to a device maker? Let’s say you want to make a device that has an LED, and you want to turn the LED on or off remotely. If your device follows the guidelines detailed by the On/Off Cluster, then your device can be controlled by the exact same ZigBee enabled wall switch discussed above. Let’s take that one step further. Replace the ZigBee enabled wall switch with a smartphone application. Say you have an existing home automation solution with a smartphone application that controls your ZigBee enabled lights (like SmartThings). Since your LED device adheres to the ZigBee On/Off cluster commands, it is now possible to control the LED by that same smartphone app! Your device just joined the Internet of Things!

You have all the power of that slick smartphone app without writing one line of custom code for your smartphone! Not only can you control your LED from your smartphone, it can also be part of your home automation. So, if you have your home automation rules set up to turn on your lights when you pull into your garage, your LED can come on as well.

Controlling a Custom Device with SmartThings

This project is going to focus on the radio hardware setup and communications required to talk to a ZigBee HA network hub (SmartThings) over ZigBee standard Clusters. We will use a Propeller microcontroller connected to a Digi XBee ZB radio to communicate with the hub. The circuit is simple and you can use any number of existing Propeller prototype boards.

will have to put a router in to relay the packets and boost the signal. Most mains powered ZigBee devices can be a router.

A router can be something as simple as a Hue light bulb. Once you add the bulb to your network, it will figure out who its neighbors are and start routing packets to them. In fact, that is how we communicate with our chicken coop door controller. We have ZigBee Hue bulbs in our landscaping that carry the traffic to our chicken coop in the back yard.

Strong Application Layer Allowing Devices to Find Each Other and Communicate

In addition to ZigBee’s strong networking layer, it also has a well-defined application layer that describes how common devices can discover each other and communicate. This is what makes ZigBee so attractive for device makers. A great deal of the common actions most devices require are categorized and defined in “cluster” definitions. ZigBee clusters are given 16-bit numbers, often

Option 1) ZB-LED PCB, Propeller Mini, and Parts	URL to buy one	Price
Order Zb-LED PCB for Prop Mini	<i>Nuts & Volts</i> Webstore	\$7
Propeller Mini	www.parallax.com/product/32150	\$25
Digi shopping cart for above board	www.digikey.com/short/t8nn8d	\$27
XBee USB Program Board	www.parallax.com/product/32400	\$25
SmartThings Hub	https://SmartThings.com	\$99

Table 2.

Option 2) Parts only (no Propeller)	URL to buy one	Price
Digi shopping cart parts list	www.digikey.com/short/t8nn8d	\$27
XBee USB Program Board	www.parallax.com/product/32400	\$25
SmartThings Hub	https://SmartThings.com	\$99

Table 3.

To play along, you will need some hardware. Here are a couple of options: Option 1) is a custom PCB and a link to the parts you can order to populate it (requires soldering); Option 2) is a list of parts you can order to work with one of your existing Propeller project boards (refer to **Tables 2** and **3**). There is an

additional parts diagram for the PCB at the article link.

Step 1) Configure Your XBee to Join a ZigBee Home Automation Network.

Each ZigBee profile has stringent security requirements that must be met before a device is allowed to join a network. Your SmartThings hub uses the ZigBee Home Automation profile security model, so we will need to configure our radio to meet those specs. To make a ZigBee network secure, each network generates its own unique network encryption key used to encrypt network packets. Your device must have this key to communicate, and that is all taken care of during the network “join” process.

The ZigBee coordinator’s (SmartThings hub) primary role during a join is to give your radio the network key and allow it to join the network. To get the network encryption key from the coordinator, our radio must be configured with the proper trust center link key. Think of it as a password you must give the coordinator before it will give you the network encryption key.

To configure your XBee radio, you will need to plug the XBee into your computer with the XBee USB adapter board. If you haven’t done so, assemble your XBee USB adapter board and Load Digi’s XCTU software. You can download it from www.digi.com/products/xbee-rf-solutions/xctu-software/xctu. You will also need to download the XBee’s configuration file

FIGURE 4.



FIGURE 5.

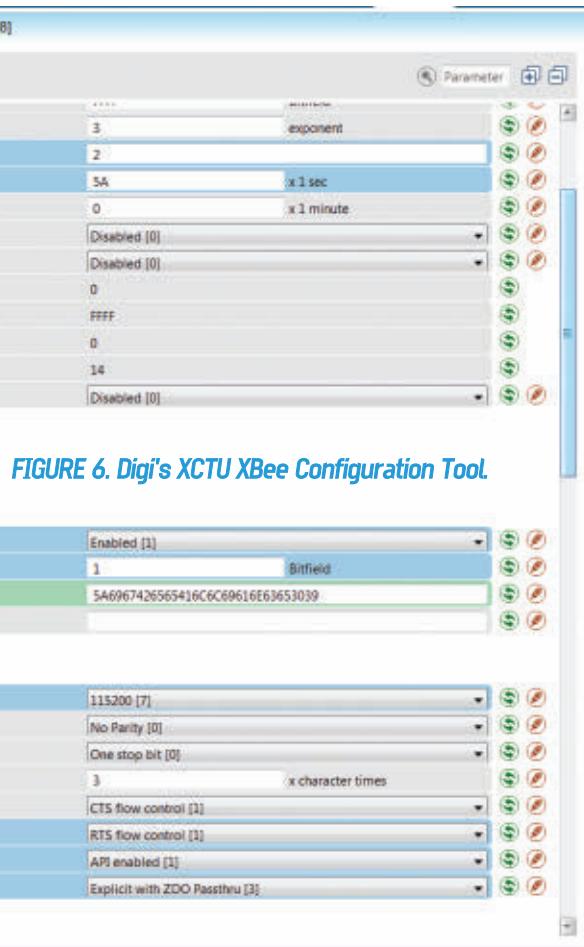


FIGURE 6. Digi's XCTU XBee Configuration Tool.

(*Nuts&Volts_xBee.xml*) available at the article link, and save it to your local hard drive. Using the XCTU software, load the configuration file from your local hard drive and write it to your XBee. (Refer to **Figures 4, 5, and 6.**) This configuration file sets the parameters shown in **Table 4**.

Once this step is complete, your radio is programmed to join any ZigBee HA network; you won't have to do this

again. As soon as you reset your radio, it will start looking for an open ZigBee HA network to join. To allow it to join the network, you will need to tell SmartThings to open the network and allow a new device to join. (More on this later. For now, you can unplug your radio and move on to the next step.)

CoopBoss ZigBee Configuration.

The following is a list of the endpoints and clusters used by the CoopBoss.

ZigBee Custom Attributes			
\$0400	Custom Attribute for Current Light RCTime value, type 0x23		
\$0401	RCTime value that will trigger a door close, type 0x23		
\$0402	RCTime value that will trigger a door open, type 0x23		
\$0403	Read Auto Door Close Setting 0=Disabled, 1=Enabled, type 0x10		
\$0404	Read Auto Door Open Setting 0=Disabled, 1=Enabled, type 0x10		
\$0405	Current of last door close		
\$0406	Seconds to next close window		
\$0407	Seconds to next open window		
\$0408	Object detection sensitivity (1 to 100) 1 = very sensitive		
\$0409	Read Normal Door Current Setting		

ZigBee End Points			
0x38	Input Cluster	0x0000	Basic Cluster
	Output Cluster	0x0101	Door Lock Cluster
	Output Cluster	0x0402	Temperature Cluster (Temperature of XBee radio)
0x39	Input Cluster	None	N/A
	Output Cluster	0x0402	Temperature Cluster (Coop Probe 1)
0x40	Input Cluster	None	N/A
	Output Cluster	0x0402	Temperature Cluster (Coop Probe 2)

ZigBee Cluster 0x0101 Attributes			
\$0000	Read LockState; see page 344 of ZCL, type 0x30	\$30 = eight-bit enumeration	
\$0001	Read LockType; see page 344 of ZCL, type 0x30		
\$0002	ActuatorEnabled; see page 344 of ZCL, type 0x10		
\$0003	Read DoorState; see page 344 of ZCL, type 0x30, Reportable		

ZigBee Custom Commands for Cluster 0x0101			
\$0A	Enable Auto Close Door	Close Door	
\$0B	Disable Auto Close Door	Open Door	
\$0C	Enable Auto Open Door	Toggle Door Lock	
\$0D	Disable Auto Open Door	High Current Door Close	
ZigBee Cluster 0x0101 commands			
\$00	Close Door		
\$01	Open Door		
\$03	Toggle Door Lock		
\$04	High Current Door Close		

Step 2) Build Your Circuit.

The sample code outlined in the next step is designed to work with the schematic in **Figure 7**. If you're building your own circuit, you don't have to worry about connecting your XBee, LED, and pushbutton to the same pin numbers shown here. We can configure that in the code.

Please note, we connect the Propeller to the XBee with five pins: Dout, Din, Reset, CTS, and RTS. We need all five pins as we are going to implement hardware handshaking to the XBee. If we had to, we could get by without the XBee's "reset" pin, but without it you may find yourself power cycling your circuit from time to time.

On pin 12 of the Propeller Mini, we connect an LED and a 220 ohm current-limiting resistor. This is the LED we will control with SmartThings.

On pin 13 of the Propeller Mini, we connect a 10K ohm pull-up resistor and pushbutton to ground. This button will also be

SmartThings awarded the CoopBoss Best in Show for their "Show Us Your SmartThings" contest. Check it out at <http://blog.smarthings.com/stories/a-smart-chicken-coop>.

Parallax features the technology used by the CoopBoss on their "learn website" at <http://learn.parallax.com/inspiration/coop-boss>.

The CoopBoss is an open source project. More information is available at CoopBoss.com. This fall, you will be able to order a complete CoopBoss system from DoodleDooCoops.com.

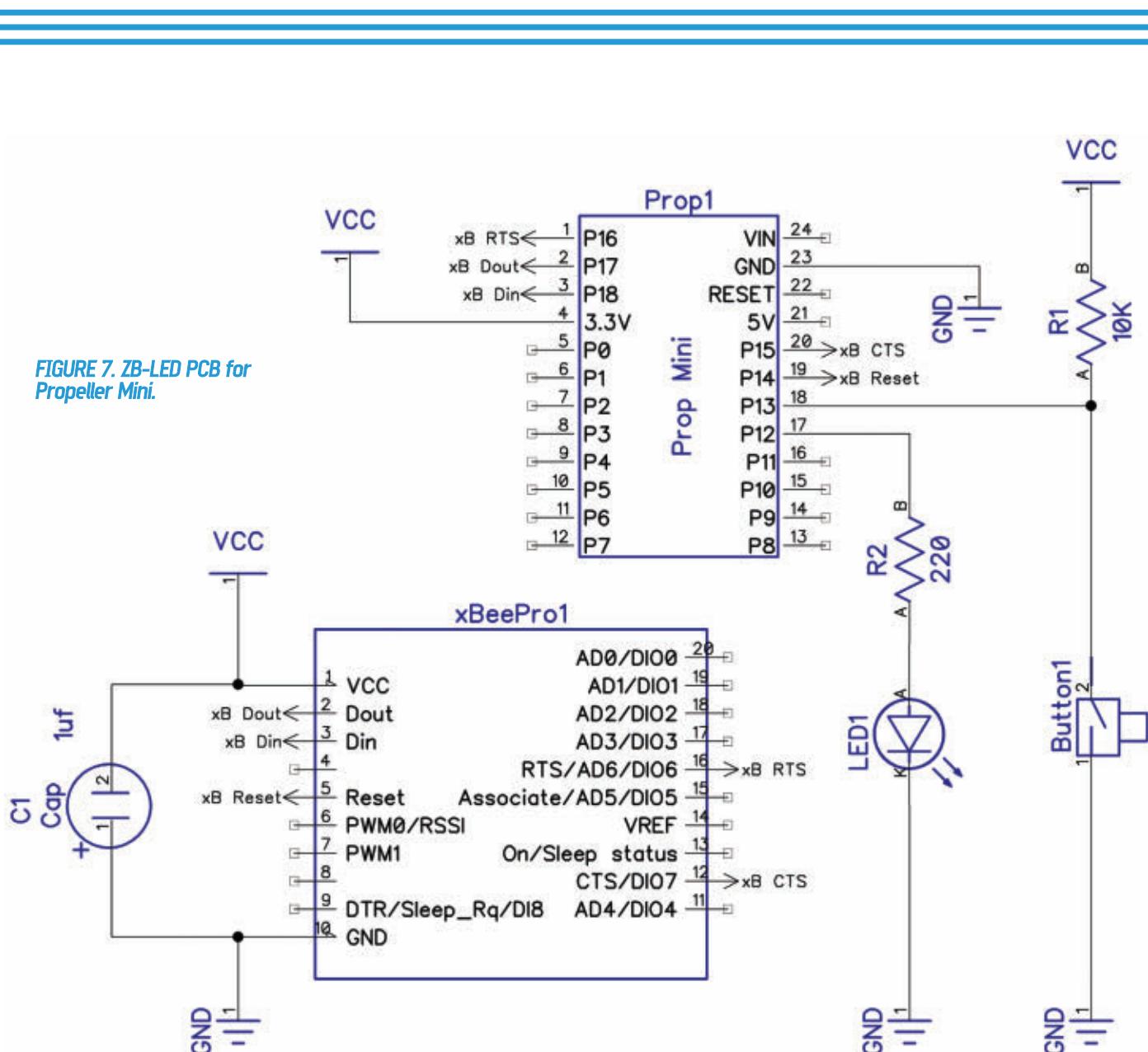


FIGURE 7. ZB-LED PCB for Propeller Mini.

used to control the LED and send commands to SmartThings.

Now that you have your circuit built, you can transfer your XBee Pro from the USB programmer to your new circuit and continue on to the next section about loading the firmware. If you're using the ZB-LED PCB (Figure 8), make sure you plug in your XBee as shown.

Step 3) Loading the Propeller Firmware.

Load the *zB_LedDemoA2.Spin* program from the article link and configure the pin

assignments for your circuit. If you're using the ZB-LED PCB, the pin assignments will be correct (Figure 9).

Once you have the pins set correctly, save the settings and then load the program (*zB_LedDemoA2.SPIN*, also at the article link) to the Propeller's EEPROM. Bring up the

Cmd	Setting	Description
ZS	0x2	Set ZigBee stack to 2
NJ	0x5A	Set network join time to 90 seconds
EE	0x1	Enable the use of encryption keys
EO	0x1	1 = Use trust center to get network key
KY	0x5A6967426565416C6C69616E63653039	This is the trust center link key
BD	0x7	Set baud rate to 115200
D6	0x1	Turn on RTS flow control
AP	0x1	Enable API mode
AO	0x3	Explicit API mode with ZDO pass-through

Table 4.

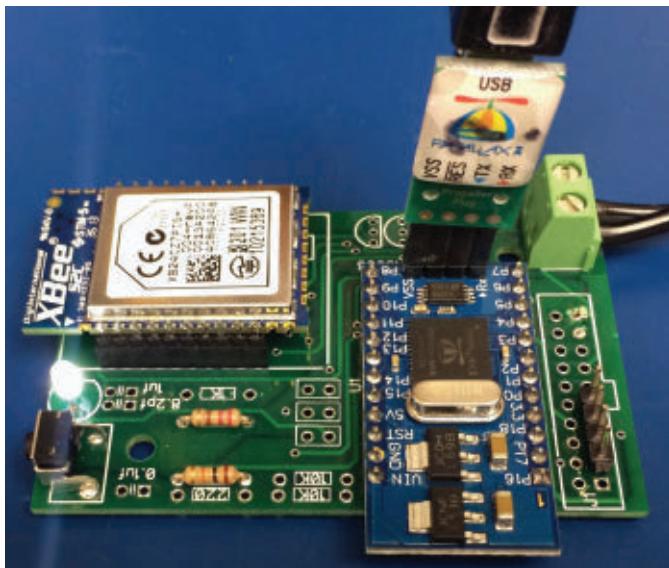


FIGURE 8.

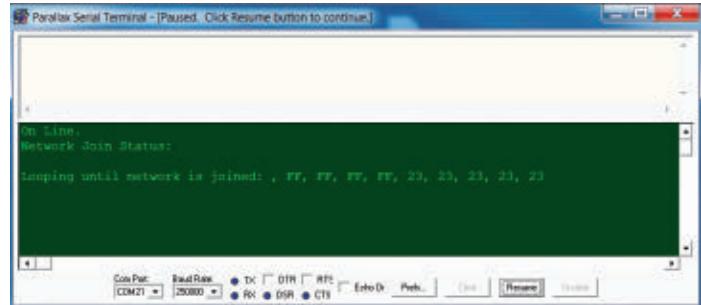


FIGURE 10.

Pin assignments:
 xBeeTx=18 . Output pin is connected to D10 pin of xBee
 xBeeRx=17 . Input pin is connected to Dout pin of xBee
 xBeeReset=14 . Output pin is connected to Reset Pin of xBee. Low for > 26ms to reset xBee
 xBeeCTS=15 . Input used for RS232. Low signal on this pin will also go high when xBee is sleeping
 xBeeRTS=16 . Output. RTS must be set low to tell xBee it is okay to send data to propeller.
 xBeeSleep=6 . Output. xBee's Sleep Request Pin. High = allow sleep, High to Low = Wake xBee. xBee will stay awake as long as pin is low
 LEDpin=12 . LED Pin
 UserBTN=13 . Push button on circuit board

Parallax Serial Terminal and cycle the power to your circuit. You should see the Propeller boot up and start looking for a ZigBee Home Automation network to join as

FIGURE 9.

shown in **Figure 10**. The two letter hex codes are what the XBee's AI (Association Indication) parameter is reporting. **Table 5** is from page 148 of the XBee/XBee-PRO ZB SMT RF Modules documentation and it details the two letter hex codes. What you should see are the hex numbers 0xFF and 0x23 repeated over and over.

The XBee radio will look for a network to join that matches its security settings. Every time you

see "FF" displayed on the screen, it is searching on a new channel; "23" means it found a valid coordinator but it is not allowing it to join at this time. To allow our device to join, we have to open the network for joining. Proceed to the next step to join the SmartThings network.

Table 5

AI Code	Description
0x00	Successfully formed or joined a network. (Coordinators form a network; routers and end devices join a network.)
0x21	Scan found no PA.
0x22	Scan found no valid PANs based on current SC and ID settings.
0x23	Valid coordinator or routers found, but they are not allowing joining (NJ expired).
0x24	No joinable beacons were found.
0x25	Unexpected state; node should not be attempting to join at this time.
0x27	Node joining attempt failed (typically due to incompatible security settings).
0x2A	Coordinator start attempt failed 0x2B — checking for an existing coordinator.
0x2C	Attempt to leave the network failed 0xAB — attempted to join a device that did not respond.
0xAC	Secure join error — network security key received unsecured.
0xAD	Secure join error — network security key not received.
0xAF	Secure join error — joining device does not have the right preconfigured link key.
0xFF	Scanning for a ZigBee network (routers and end devices).

Step 4) Connect to SmartThings.

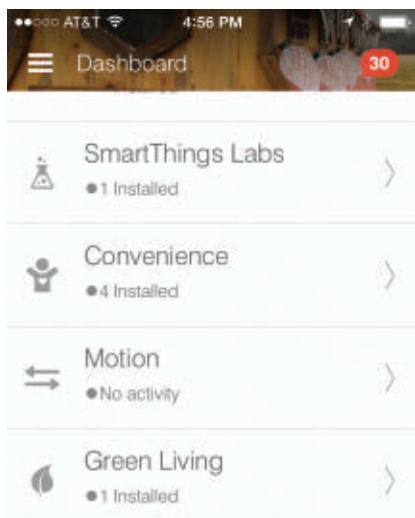
If you haven't set up your SmartThings hub and downloaded their smartphone app, you need to do that now.

From your smartphone, open the SmartThings app (**Figure 11**) and go to the Marketplace; scroll until you see "Connect New Device" as shown in **Figure 12**. Tap on "Connect New Device" to tell SmartThings to open the network and let new devices join. After a few minutes, you should see the WeMo Bulb pop up on the bottom part of your smartphone's screen (**Figure 12**). We are impersonating that device and you can tap to configure it and change its name to "My LED" or just tap "Done" to accept the defaults.

Back on your Parallax Serial Terminal screen, you will notice that ZigBee packets are starting to flow as shown in **Figure 13**. Your device has joined the SmartThings Home Automation network and now can be controlled with your smartphone.

Find your WeMo Bulb or, in our case, "My LED" in your things listing from the dashboard, and tap on the icon to turn the LED on and off.

FIGURE 11.



Bang! Your LED will turn on and off as you push the button (**Figures 14** and **15**). Since SmartThings is cloud-based, you will be able to take your smartphone anywhere you have cell phone or Wi-Fi signal and control your LED.

In a follow-up article, we will walk through the code details of what's going on behind the scenes and add more functionality to our device by adding support for the Level Cluster that will allow us to dim the LED, and ZigBee binding that will allow us to push the button and change the status of our LED in SmartThings in real time. **NV**

Parallel Serial Terminal (Paused. Click Resume button to continue.)

FIGURE 13.

On Line.
Network Join Status:

Joining until network is joined: , FF, FF, FF, FF, 23, 23, 23, 23, 23, FF, 23, 23, 00
IEEE Addr 10A300-40R543F8
Net Addr 0082

Device Announcement Just Ready.
Send packet received:
 IEEE Addr: 0052A8003C8E0001
 Net Addr: 0000
 Dst EP: 00
 Src EP: 00
 Profile ID: 0000
 Cluster ID: 0005

3 bytes → 02 82 07 ←
Reporting Active End Points.
Send packet received:
 IEEE Addr: 0052A8003C8E0001
 Net Addr: 0000
 Dst EP: 00
 Src EP: 00
 Profile ID: 0005
 Cluster ID: 0004

4 bytes → 03 82 07 38 ←
Reporting Simple Desc for End Point 38..
Send packet received:
 IEEE Addr: 0052A8003C8E0001
 Net Addr: 0000
 Dst EP: 01
 Src EP: 01
 Profile ID: 0104
 Cluster ID: 0006

11 bytes → 06 00 06 00 00 00 10 00 00 10 9E ←

Switch off.
Send packet received:
 IEEE Addr: 0052A8003C8E0001
 Net Addr: 0000
 Dst EP: 01
 Src EP: 01
 Profile ID: 0104

FIGURE 12.

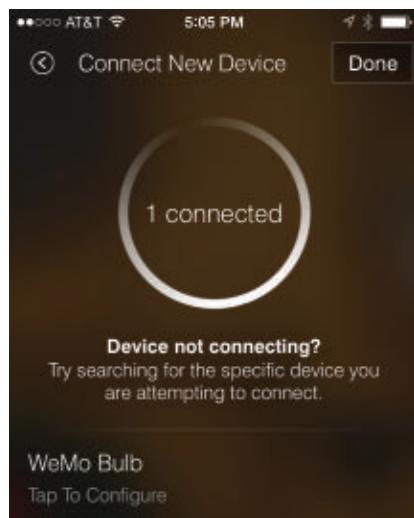


FIGURE 14.

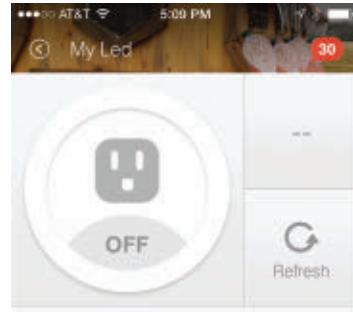
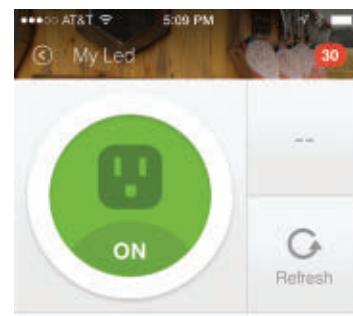


FIGURE 15



RF Specialists

"Making your RF ideas into profitable products."

FCC Part 90 Compliant

USX2 - NBFM Multi-Channel UHF Transceiver with Programmable RF Power



MURS

(Multi-Use Radio Service)
SHX1 - Long Range, High Power MURS Band Transceiver



ZigBee Pro

OEM Modules and USB ZigBee Sticks, Mesh Networks




Industrial Bluetooth

OEM, Modules, Wireless Device Servers, RS-232 Long range options, low cost



Ultra-Low Power Wi-Fi networking module and Eval board

The AMW006 'Numbat' module is an ultra-low power Wi-Fi networking module with full regulatory certification.



RF Design Services

Prepared to work with your in-house engineers, or support your RF project from initial design to implementation.

Industrial • Military • Space • Medical • Smart Grid Metering • SCADA • Lighting Control



LEMOS

INTERNATIONAL

Tel: 1.866.345.3667
orders@lemosint.com
www.lemosint.com

ELECTRONET

USB

Add USB to your next project--
It's easier than you might think!



USB-FIFO • USB-UART • USB/Microcontroller Boards
RFID Readers • Design/Manufacturing Services Available
Absolutely NO driver software development required!

www.dlpdesign.com

For the ElectroNet

online, go to

www.nutsvolts.com

click **Electro-Net**

ALL ELECTRONICS
CORPORATION

Electronic Parts
and Supplies.

www.allelectronics.com

Free 96 page catalog 1-800-826-5432

INVEST in your BOT!



12115 Paine Street • Poway, CA 92064 • 858-748-6948 • www.hitecrcd.com

CORIDIUM
Floating point **BASIC** for
ARM controllers from \$5.00
www.coridium.us

PCB,PCBA
and More!
Myro
Low cost
High Quality
www.myropcb.com
1-888-PCB-MYRO

Ironwood
ELECTRONICS
For Any PCB-GHz
Bandwidth BGA/QFN Sockets
Quick-Turn Custom Sockets
www.ironwoodelectronics.com

superbrightleds.com
COMPONENT LEDs • LED BULBS • LED ACCENT LIGHTS

MANTIS
six-wheel drive
TAKE YOUR OFF-ROAD EXCURSIONS TO
THE NEXT LEVEL! SERVOCITY.COM

**Did You Know Preferred
Subscribers get access to all the
digital back issues of
Nuts & Volts for free?
Call for details
1-877-525-2539**

www.servomagazine.com

MOBILE APP NOW AVAILABLE!

SERVO

Download NOW On Your
Favorite Mobile Device!

**FOR
ROBOT
BUILDERS**

iOS • ANDROID • KINDLE FIRE

Did you know that *Nuts & Volts*
now has a weekly content
newsletter? **Want to get it?**

You have three ways to sign up:

- Visit us on Facebook and click on "Join My List."
- Using your cell phone, send "NVNEWSLETTER" as a text message to 22828.
- Visit the *Nuts & Volts* FAQs to sign up at nutsvolts.com/faqs

Vintage Computing

Five Easy Projects for Your Mentor's Friend

By Dane Weston

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?/magazine/article/january2016_RetroComputer_Amigo.

As described in last month's issue of Nuts & Volts, the Mentor's Friend is a retro computer that you build yourself and program in BASIC. It is easy to build and fun to play with, and it offers a superb technology mentoring platform you and your child (or grandchild) can build and use together to explore computers and programming.

Once you've built yours, here are five easy projects to get you and a young one started on a technology adventure. These have been selected to be engaging, and to give you and your student a hands-on introduction to different components of Color BASIC. For reference, you may find it helpful to download the "Intro to Color BASIC Commands" from the article link. Have fun!

1. KEYNUM.BAS.

Color BASIC often takes user input from the keyboard as individual keystrokes, with each key represented by an assigned integer number. So it's important to know what number represents each key. This first program does that translation for you, and as a bonus shows you the character glyph for any key press.

To input the program, type NEW <Enter> at the flashing rectangle cursor on your Amigo, then press <F1> on your keyboard to enter the Color BASIC editor. Next, type in the code that follows; press <Enter> at the end of each line. In the editor, you can use the arrow, <Home>, and <End> keys to navigate, and the <Backspace> and <Delete> keys to erase text.

When you've finished typing in the code, press <F1> again to return to Color BASIC and run the program. Don't worry if you see a syntax error — just use the editor again to find and correct the offending line.

```

10 REM *** KEYNUM.BAS ***
20 COLOR 63,22
30 CLS
40 LOCATE 10,0: PRINT ">>> Keycode Values <<<"
50 LOCATE 1,2: PRINT "Press Any Key, or <CTRL> Q
   to Return..."
60 REM -- Get Keystroke --
70 A=INKEY
80 IF A=0 THEN GOTO 70 REM <- No Key Pressed
90 REM -- Show Value and Glyph --
100 LOCATE 18,18: PRINT "INKEY = ";
110 PRINT A,
120 LOCATE 16,20: PRINT "DISPLAY = ";
130 DISPLAY A
140 REM -- Again? --
150 IF A<>593 AND A<>625 THEN GOTO 70 REM <-
   CTRL Q?
160 CLS: END

```

After you've explored the keyboard for a bit, press <CTRL-Q> to end the program. (In Color BASIC, you can also press <Esc> at any time to interrupt program flow.) Then, enter SAVE "KEYNUM.BAS" at the prompt. That will save your program to the SD card, so you can LOAD "KEYNUM.BAS" and RUN it when you wish. Remember that you can type DIR at the prompt to see a directory of the files on the SD card, now including KEYNUM.BAS (**Figure 1**).

Here's what's going on in this program:

- Lines 10, 60, 90, and 140 use the REM command (for Remark) to provide some internal documentation to explain what's going on. It is good programming practice

to provide "enough" documentation in your code. Also note the REM at the end of Line 150 – some (but not all!) Color BASIC commands allow a following REM on the same program line. Color BASIC ignores everything on a program line after the REM.

- Lines 20-50 set the screen color to white text on a blue background, clear the screen, and print the app title and user instructions. The LOCATE command positions the cursor at a specified column and row, where column is between 0 and 49 and row is between 0 and 36, inclusive. Note the use of the colon in lines 40 and 50 to place more than one Color BASIC command on the same program line.

- Line 70 uses the INKEY command to get an integer value from the keyboard buffer and store it in variable A.

- Line 80 checks the value in A. If it is 0, no key has been pressed, and program flow jumps back to line 70 to check again. If it is not 0, program flow continues at line 90.

- Lines 100-130 print the assigned integer and character glyph of the keystroke value in variable A. The semicolon at the end of lines 100 and 120 keeps the cursor positioned immediately after the printed text (no line feed/carriage return). The comma at the end of line 110 appends a tab character to ensure previous entries are completely erased. Note that PRINT A shows the integer in variable A, while DISPLAY A shows the assigned character glyph.

- Line 150 checks to see whether the keystroke was a CTRL Q or CTRL q. If it wasn't, program flow jumps back to line 70 to wait for another key press. If it was, program flow continues at the next line.

- Line 160 clears the screen to indicate the program end to the user, then ends the program. The END command stops program execution and returns the Amigo to the "immediate mode" flashing rectangle prompt. Since Color BASIC stops program execution automatically when it runs out of program lines, END is not required but is considered good programming practice.

2. COLORS.BAS.

Sometimes you need to find just the right foreground and background colors for the screen text in your programs. This next program makes that process easy. Fire up the editor (press <F1>) to enter and run (<F1> again) this code:

```

10 REM *** COLORS.BAS ***
15 COLOR 63,0
20 CLS
25 LOCATE 15,0: PRINT ">>> Color Values
   <<<"
30 PRINT " "
35 REM -- FOR/NEXT Loop Shows Colors --

```

FIGURE 1: Knowing keycode values is helpful in Color BASIC.

```

40 FOR C=0 TO 63
45 COLOR C,0
50 PRINT C,
55 NEXT C
60 REM -- Get Colors; Print Sample --
65 COLOR 63,0
70 PRINT " "
75 INPUT "Input Foreground, Background Numbers:
    ";F,B
80 COLOR F,B
85 PRINT " Sample Text "
90 REM -- Another Sample? --
95 COLOR 63,0
100 PRINT " "
105 PRINT "More (Y/N)?"
110 A=INKEY
115 IF A="Y" OR A="y" THEN GOTO 65
120 IF A<>"N" AND A<>"n" THEN GOTO 110
125 END

```

When you quit this program, don't forget to copy it to the SD card with SAVE "COLORS.BAS." Here's what's going on in this program:

- Lines 10-30 clear the screen and print the title in white on black.
- Lines 35-55 print each number from 0 to 63 in its assigned Color BASIC color. The FOR...NEXT loop starts variable C at a value of 0 on the first loop, then increments C by 1 on subsequent loops until C equals 63.
- On each iteration of the loop, line 45 changes the foreground color to the current value of C, while keeping the background 0 (black).
- Line 50 prints the current value of C in that current color combination, followed by a Tab character. (The comma after a PRINT statement tells Color BASIC to append a horizontal tab instead of a new line to whatever was just printed.)
- Lines 60-85 prompt the user to input values for foreground and background colors to sample, and stores those values in variables F and B.
- Line 75 is the INPUT statement which has the syntax INPUT "<text prompt>"; <variable1>, <variable2>,... INPUT is the command to use in your programs when you need to get an integer value from the user. The text prompt (if used) is enclosed in quotes, and a semicolon must separate the prompt from the variables. Multiple variables can be input if separated by commas. If the number of variables input by the user differs from the number expected in the statement, Color BASIC returns a syntax error.
- Line 80 sets the user-selected foreground and background colors.
- Line 85 prints a text sample in this color set.
- Lines 90-125 do a yes-no check on whether the

user wants to try another sample, then direct program flow accordingly.

- Lines 95-105 print the user prompt in white on black.
- Line 110 gets a keystroke value from the keyboard and stores it in variable A. Each key is assigned a unique number value, which can be retrieved by the INKEY command after that key is pressed. If no key has been pressed, INKEY returns a value of zero.
- Line 115 checks to see whether the value in variable A is "Y" or "y." If it is, program flow jumps to line 65 to get another user input.
- Line 120 checks to see whether A is "N" or "n." If it is, program flow continues at line 125. If it is not, program flow jumps back to line 110 to get another keypress. Note that this latter branch includes the case where no key has been pressed (A=0).
- Line 125 marks the formal end of the program (Figure 2).

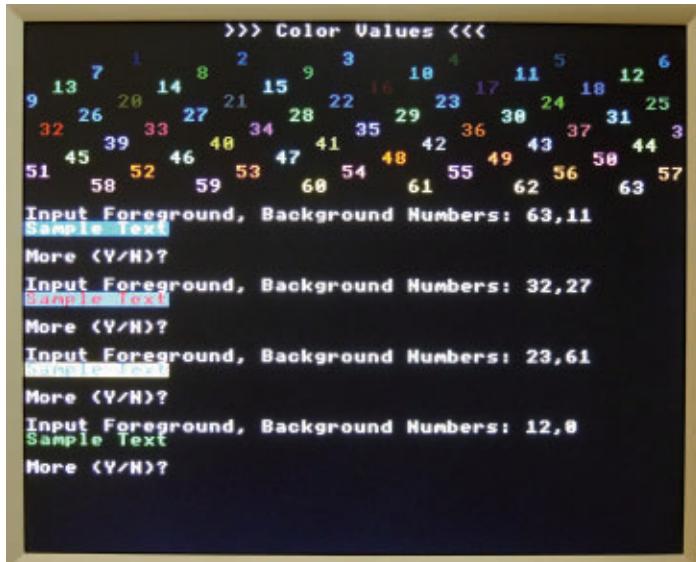


FIGURE 2: The right color combinations can dress up your screens.

3. MENU.BAS.

As you learn Color BASIC, you'll probably collect several programs that you run frequently. This program gives you a "Launchpad" for convenient access to those. Use the editor to enter and run this code:

```

10 REM *** MENU.BAS ***
20 REM -- Print Menu Screen --
30 COLOR 63,19
40 CLS
50 PRINT " " : PRINT " >>> My Amigo Launchpad
      <<<": PRINT " "

```

```

60 PRINT " F1 - Quit Launchpad"
70 PRINT " F2 - Screen Colors"
80 PRINT " F3 - Keyboard Numbers"
180 REM - Add F4 - F12 Menu Item Print
    Statements Above
190 PRINT " :: PRINT " >>> Press Function Key to
    Launch App"
200 REM -- Get Keyboard Input --
210 k=INKEY
220 IF k<208 OR k>219 THEN GOTO 210
230 REM -- RUN Selected App --
240 GOTO 250 + (k-208)
250 CLS: END           REM - F1
251 RUN "COLORS.BAS"   REM - F2
252 RUN "KEYNUM.BAS"   REM - F3
253 REM - F4
254 REM - F5
255 REM - F6
256 REM - F7
257 REM - F8
258 REM - F9
259 REM - F10
260 REM - F11
261 REM - F12
270 GOTO 210

```

When you quit the program, don't forget to copy it to the SD card with SAVE "MENU.BAS." Here's what's going on in the program:

- Lines 10-190 print the menu screen in white on a blue background. Line numbers 80-160 are not used in order to make room for future menu expansion for keys F3-F11.
- Lines 210-220 check for keystrokes until they find one between 208 and 219 (F1 to F12), then continue the program at line 230.
- Line 240 directs program flow to a specific line number, based on which function key was pressed. The GOTO command in Color BASIC can direct program flow to the result of an expression, as well as an explicit line number. Since the keystroke value in variable k is between 208 and 219 (inclusive), the expression 250 + (k-208) in line 240 evaluates to 250 to 261, and the GOTO directs program flow to the line number assigned to the selected function key.
- Line 250 is the option to exit the program. It clears the screen and ends the program.
- Line 251 instructs Color BASIC to load and run the COLORS.BAS program. Doing this removes MENU.BAS from program memory; we'll need to make a minor change to COLORS.BAS (discussed later) to automatically return to the Launchpad program.
- Line 252 loads and runs KEYNUM.BAS. Again, we'll need to make a minor change to KEYNUM.BAS to get it

to restart the Launchpad.

- Lines 253-261 are stubs where you can insert additional RUN commands for keys F4-F12 to launch other programs. Add them in order, starting with F4.
- Line 270 sends program flow back to get another keystroke in case an undefined function key is pressed (**Figure 3**).



FIGURE 3: Add your own programs to MENU.BAS to customize your Launchpad.

As mentioned earlier, getting the target programs to restart the Launchpad app requires a slight change to each target program. For COLORS.BAS, replace the END in line 125 with RUN "MENU.BAS." For KEYNUM.BAS, replace the CLS: END in Line 160 with RUN "MENU.BAS." Do this for any additional programs you create and add to your Launchpad app.

I hope your Amigo Launchpad will help showcase your programming projects to family and friends!

4. FACES.BAS.

This little program exercises a couple of graphics commands and the random number generator to produce some artwork on the monitor. Use the editor to enter and run the code below.

You should see a piece titled "Faces in the Crowd," and it won't take you long to understand the name. Don't forget to SAVE "FACES.BAS" after you've enjoyed the Amigo art for a bit.

```

5 REM *** Faces in the Crowd ***
10 REM ... Your Amigo Does Art...
15 REDEFINE 33,60,66,165,129,165,153,66,60
20 COLOR 63,0
25 CLS

```

```

30 X=RND (100)
35 Y=RND (75)
40 C=RND (64)
45 PLOT X,Y,C
50 COLOR 63,0
55 LOCATE 15,1: PRINT " Faces in the Crowd ";
60 PAUSE 100
65 IF X<>50 THEN GOTO 30
70 COLOR C,0
75 LOCATE RND (50),RND (37)
80 DISPLAY 33
85 GOTO 30

```

Here's what's going on in this little program:

- Lines 5 and 10 provide a tiny bit of internal documentation.
- Line 15 redefines the exclamation point character (ASCII 33) for artistic use in our program. Each of the eight numbers following the 33 (and separated by commas) is the decimal value of one binary row of on-or-off pixels in the 8x8 pixel map for that character. These values (double-check when you enter them) will change the exclamation point character to a smiley face to give our artwork some punch. (A little REDEFINE.BAS program is included on the SD card in the Amigo kit from the *Nuts & Volts* webstore – check it out!)
- Lines 20 and 25 set a black background and clear the screen.
- The Color BASIC screen has 100 graphics columns numbered 0 to 99, and 75 graphics rows numbered 0 to 74. The PLOT command has the syntax PLOT <column>, <row>, <color>. Lines 30, 35, and 40 pick random integers for column, row, and color, and assign those values to variables X, Y, and Z.
- Line 45 plots a small square (one quarter of a text cell) at the randomly chosen graphics column and row, in the randomly chosen color.
- Lines 50 and 55 set the color to white on black, and prints the title of our artwork. We reprint the title every cycle of the program, just in case one of the random squares lands on our title.
- Line 60 pauses the program for 100 milliseconds to slow things down a bit for a good visual effect. You can adjust this value to your preference.
- Line 65 directs program flow back to line 30 to plot another random square, unless the value for X on that cycle is 50. In that case, program execution continues at line 70. (The value of 50 here is arbitrary, and could have been any permissible value for X.) You can set a different condition after the IF to plot more (or fewer) smiley faces.
- Line 70 sets the foreground color to the random value previously chosen for C, keeping the background color black.
- Line 75 moves the cursor to a random text column

and row. We could have assigned the random column/row values to variables first (A=RND (50): B=RND (37): LOCATE A,B), but we're not reusing the values. So, this way works just fine, and doesn't tie up any of our Color BASIC variables.

- Line 80 displays the character whose assigned ASCII value is the number that follows. Since we redefined ASCII character 33 from the exclamation point to a smiley in line 15, DISPLAY 33 prints a smiley in the current foreground color at the cursor location (we could also achieve the same thing here with PRINT "!"). Multiple characters can be included in the same DISPLAY command, like DISPLAY 65,66,67.

- Line 85 directs program flow back to line 30 to plot another random square (and perhaps a smiley). Since the program has no end, it will remain in this endless loop until interrupted with the <Esc> key. Enjoy your Amigo artwork (**Figure 4**)!



FIGURE 4: Your Amigo does art?

5. SOUNDfx.BAS.

This program takes keystrokes from the keyboard and uses them to drive the audio channel. You'll need to connect a small powered speaker to your Amigo audio jack to hear the results. (Earbuds will do in a pinch.) Use the editor (<F1>) to enter and run (<F1> again) this code:

```

10 REM *** SOUNDfx.BAS ***
15 REM ... Your Amigo Makes Music? ...
20 COLOR 63,22
30 CLS
40 PRINT " SOUND FX": PRINT " "
50 PRINT " Connect earbuds to your Amigo audio
jack."

```

Building the Mentor's Friend

As discussed in the December 2015 issue of *Nuts & Volts*, the Mentor's Friend (nicknamed the Amigo) is a direct descendant of the Pocket Mini Computer (PMC) — a project by Jeff Ledger that combined the work of several Propeller gurus to offer a series of retro computers running a version of BASIC. When the PMC was no longer commercially available, Dane Weston developed a simplified replacement in mid-2015 to provide a homebuilt, BASIC-driven mentoring platform for *Nuts & Volts* readers.

Perhaps the easiest way to build your own Amigo is from the kit available from the *Nuts & Volts* webstore which provides everything you need, including the circuit board, all components (including an EEPROM pre-loaded with Color BASIC), and a 2 GB SD card with sample programs. However, if you already have a Propeller board with connectors for a PS/2 keyboard, VGA monitor, and 2 GB SD card, you can "roll your own" Amigo by downloading the Color BASIC source from the article link, updating the I/O pin assignments to match your hardware, and then loading the compiled Color BASIC binary to the EEPROM on your board. Or, if you are truly ambitious, download the Amigo schematic (also at the article link), build out the circuits using your favorite construction techniques, then download the Color BASIC source or binary and Flash it to your EEPROM.

Regardless of the method you choose, please use the N&V blogs to share your challenges and triumphs, or to ask for advice if needed. Good luck!

```

60 PRINT " Press most any key to generate a
sound."
70 PRINT " Press <Space> to stop the sound."
80 PRINT " Press CTRL Q to quit.
90 PRINT " Give it a try!"
100 a=INKEY
110 IF a=0 THEN GOTO 100
120 LET b=a
130 IF b>90 THEN b=b/2
140 NOTEON b
150 IF a=" " THEN NOTEOFF
160 IF a=593 OR a=625 THEN NOTEOFF : END
170 GOTO 100

```

Again, when you quit this program, don't forget to copy it to the SD card with SAVE "SOUNDFX.BAS." Here's what's going on in this program:

- Line 20 sets the foreground color to white and the background to blue.
- Line 30 clears the screen to the current background color (now blue).
- Lines 40-90 print the program name and instructions.

The Easiest Way to Design Custom Front Panels & Enclosures

Free
Front Panel
Designer



You design it
to your specifications using
our FREE CAD software,
Front Panel Designer

We machine it
and ship to you a
professionally finished product,
no minimum quantity required

- Cost effective prototypes and production runs with no setup charges
- Powder-coated and anodized finishes in various colors
- Select from aluminum, acrylic or provide your own material
- Standard lead time in 5 days or express manufacturing in 3 or 1 days

**FRONT PANEL
EXPRESS**

FrontPanelExpress.com

WORLD'S MOST VERSATILE CIRCUIT BOARD HOLDERS

Model 324 Our Circuit Board Holders add versatility & precision to your DIY electronics project. Solder, assemble & organize with ease.



MONTHLY CONTEST

Visit us on Facebook® to post a photo of your creative PanaVise project for a chance to win a PanaVise prize package.



PANAVISE®
Innovative Holding Solutions

7540 Colbert Drive • Reno • Nevada 89511 | (800) 759-7535 | www.PanaVise.com

- Line 100 gets a single character from the keyboard and places its integer keycode in the variable "a." Color BASIC has 26 variables – A through Z – and each is case insensitive. So, variable "A" is the same as "a." Some Color BASIC programmers use lowercase variables to make them easier to see in the uppercase commands.

- Line 110 checks the keycode in variable a. If it is zero, no key has been pressed, and program flow jumps back to line 100 to check again for a keystroke.

- Line 120 copies the value in variable a to variable "b," so we can change the copy while preserving the original value. The use of LET is optional in Color BASIC value assignment, so we could have just said b=a without the LET.

- Line 130 checks the integer value in b and – if it is greater than 90 – divides it by 2 to produce a more mid-range sound. (Higher numbers with the NOTEON command in line 140 produce hard-to-hear high pitched sounds.) Note that when playing your Amigo synthesizer, you can use the Caps Lock key to change the tone reference for the letter keys.

- Line 140 uses the processed keycode to turn on a corresponding continuous tone. The note will continue until another NOTEON or a NOTEOFF command is received. Color BASIC will only play one note at a time.

- Line 150 checks the original keycode to see whether the CTRL Q key combination (in either upper or lower case) has been pressed. If so, it turns off any current tone and ends the program. (A tone will continue after the program ends unless it is turned off with NOTEOFF.)

Dane Weston's book on *Color BASIC for the Pocket Mini Computer* is available at www.lulu.com/shop/display-product.ep?pID=4536358.

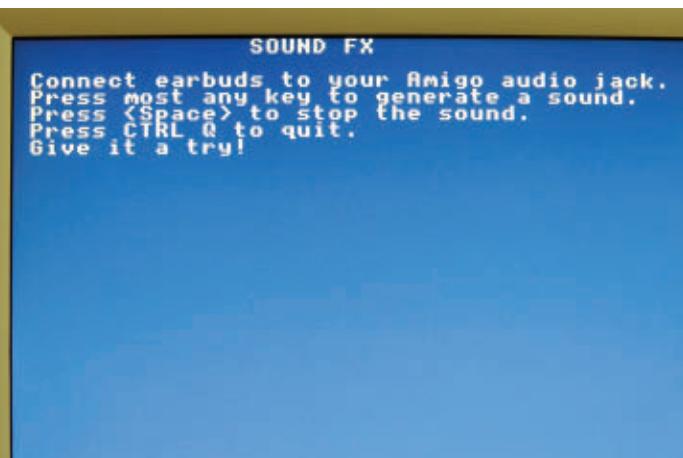


FIGURE 5: Just add earbuds for a simple sci-fi synthesizer!

- Line 160 checks the original keypress for a <Space>. If one is detected, it turns off any current tone.

- Line 170 directs program flow back to get another keystroke, and does it all again.

Enjoy your sci-fi synthesizer (Figure 5)!

That's a Wrap

This completes the five easy projects for the Mentor's Friend. I hope you and any young helpers have fun, and that these projects get you off to a good start with the Amigo and Color BASIC.

We'll do some future articles that will cover more about Color BASIC, as well as some fun hardware projects to control with the Amigo. Until then, be well, my friends!

NV

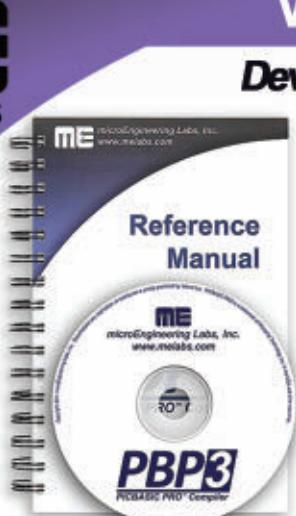


www.melabs.com

Development Tools and Learning Systems

PICMicro® microcontrollers

Buy Now!



- PICBASIC PRO™ Compiler 3.0
- M.E. Labs Trainer
- U2 Programmer
- Prototyping Boards
- Experimenter Boards



Start learning programming today with M.E. Labs trainer board. Introductory Price \$129.95

PICBASIC and PICBASIC PRO are trademarks of Microchip Technology Inc. in the USA and other countries. PIC is a registered trademark of Microchip Technology Inc. in the USA and other countries.

For complete product details, visit our webstore!

The Nuts & Volts **WEBSTORE**

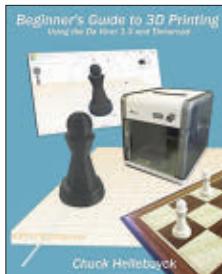
GREAT FOR DIYers!

Beginner's Guide to 3D Printing

by Chuck Hellebuyck

This book was written to answer the questions most beginners need answered. It covers many of the popular 3D printer choices and then uses the under \$500 Da Vinci 1.0 from XYZprinting to show you how easy it is to get started. I take you through using Tinkercad software for creating your own custom designs. I go further and show you how to take a simple design and send it off to a professional 3D printer. This book was designed for anyone just getting started.

\$24.95

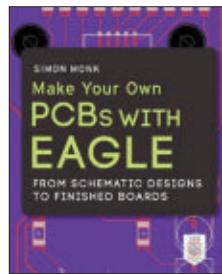


Make Your Own PCBs with EAGLE

by Eric Kleinert

Featuring detailed illustrations and step-by-step instructions, *Make Your Own PCBs with EAGLE* leads you through the process of designing a schematic and transforming it into a PCB layout. You'll then move on to fabrication via the generation of standard Gerber files for submission to a PCB manufacturing service. This practical guide offers an accessible, logical way to learn EAGLE and start producing PCBs as quickly as possible.

\$30.00



How to Diagnose and Fix Everything Electronic

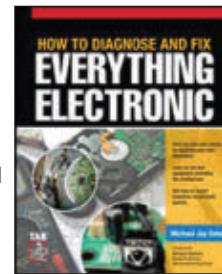
by Michael Jay Geier

Master the Art of Electronics Repair

In this hands-on guide, a lifelong electronics repair guru shares his tested techniques and invaluable insights. *How to Diagnose and Fix Everything*

Electronic shows you how to repair and extend the life of all kinds of solid-state devices, from modern digital gadgetry to cherished analog products of yesteryear.

\$24.95



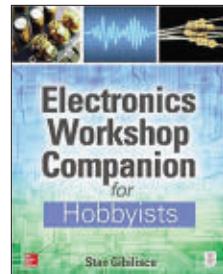
Electronics Workshop Companion for Hobbyists

by Stan Gibilisco

In this practical guide, electronics expert Stan Gibilisco shows you, step by step, how to set up a home workshop so you can invent, design, build, test, and repair electronic circuits and gadgets.

Electronics Workshop Companion for Hobbyists provides tips for constructing your workbench and stocking it with the tools, components, and test equipment you'll need. Clear illustrations and interesting do-it-yourself experiments are included throughout this hands-on resource.

\$25.00



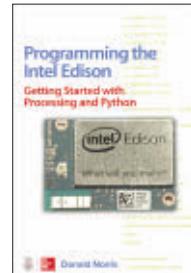
Programming the Intel Edison: Getting Started with Processing and Python

by Donald Norris

Learn To Easily Create Robotic, IoT, and Wearable Electronic Gadgets!

Discover how to set up components, connect your PC or Mac, build Python applications, and use USB, Wi-Fi, and Bluetooth connections. Start-to-finish example projects include a motor controller, home temperature system, robotic car, and wearable hospital alert sensor.

\$20.00

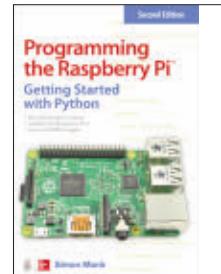


Programming the Raspberry Pi Second Edition: Getting Started with Python

by Jack Simon

This practical book has been revised to fully cover the new Raspberry Pi 2, including upgrades to the Raspbian operating system. Discover how to configure hardware and software, write Python scripts, create user-friendly GUIs, and control external electronics. DIY projects include a hangman game, RGB LED controller, digital clock, and RasPiRobot complete with an ultrasonic rangefinder.

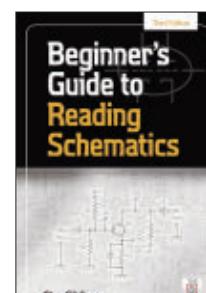
\$15.00



Beginner's Guide to Reading Schematics, 3E

by Stan Gibilisco

Navigate the roadmaps of simple electronic circuits and complex systems with help from an experienced engineer. With all-new art and demo circuits you can build, this hands-on, illustrated guide explains how to understand and create high-precision electronics diagrams. Find out how to identify parts and connections, decipher element ratings, and apply diagram-based information in your own projects.

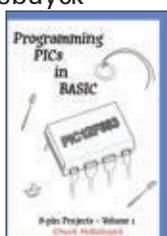


\$25.00

Programming PICs in Basic

by Chuck Hellebuyck

If you wanted to learn how to program microcontrollers, then you've found the right book! Microchip PIC microcontrollers are being designed into electronics throughout the world and none is more popular than the eight-pin version. Now the home hobbyist can create projects with these little microcontrollers using a low cost development tool called the CHIPAXE system and the Basic software language. Chuck Hellebuyck introduces how to use this development setup to build useful projects with an eight-pin PIC12F683 microcontroller.



\$14.95

Programming Arduino Next Steps: Going Further with Sketches

by Simon Monk

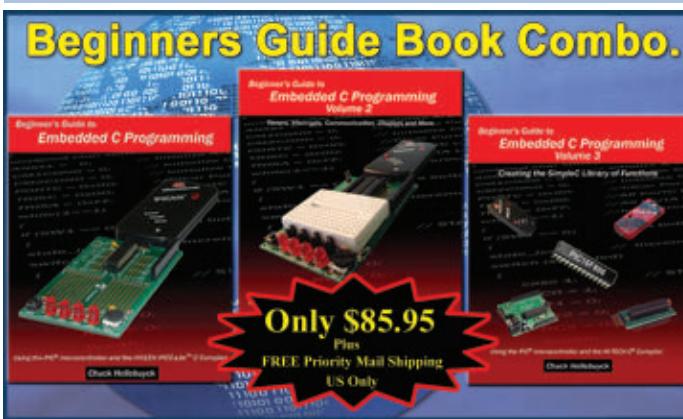
In this practical guide, electronics guru Simon Monk takes you under the hood of Arduino and reveals professional programming secrets. Also shows you how to use interrupts, manage memory, program for the Internet, maximize serial communications, perform digital signal processing, and much more. All of the 75+ example sketches featured in the book are available for download.



\$20.00

Order online @ store.nutsvolts.com
Or CALL 1-800-783-4624 today!

EDUCATIONAL



DO YOU LOVE RADIOS?

TRANSISTOR RADIOS
Sale Price \$39.95

Ham and Shortwave Radio For the Electronics Hobbyist
Sale Price \$19.95

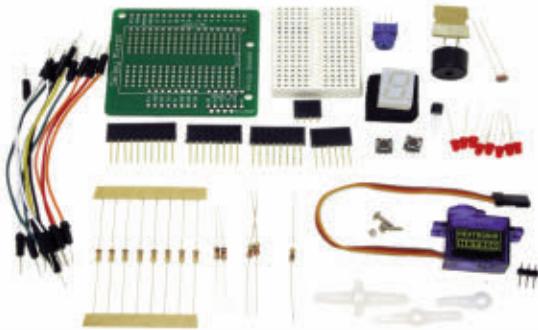
ARDUINO PROJECTS for AMATEUR RADIO
Sale Price \$23.95

Get 20% off these three books
To order call 800 783-4624 or visit: <http://store.nutsvolts.com>

Arduino Classroom - learn computing and electronics

The free Internet virtual textbook: Arduino 101 at www.arduinoclassroom.com provides a sensible learning sequence that introduces computing and electronics with clear text and detailed hands-on labs with tested examples using the Arduino Projects Kit.

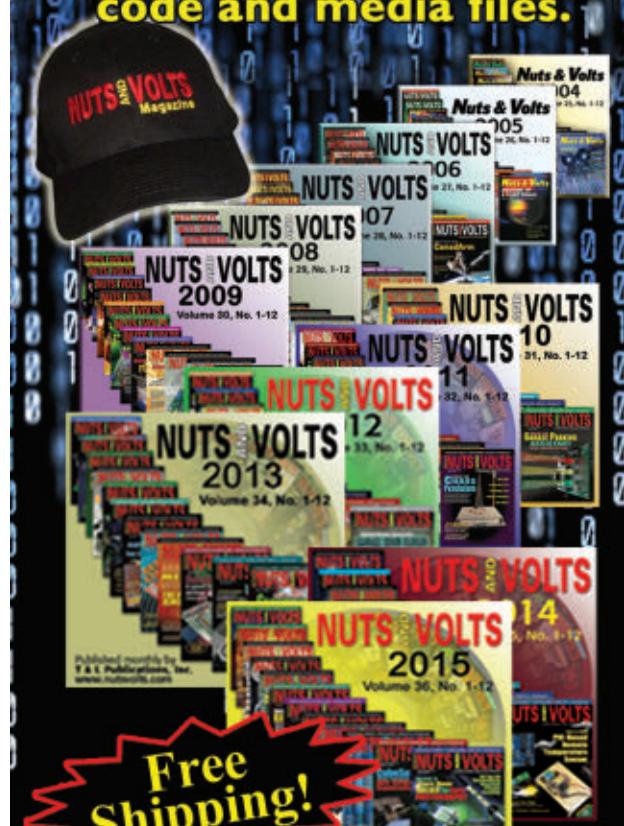
Available from Nuts&Volts for only \$44.99



CD-ROM SPECIAL

Nuts & Volts 12 CD-ROMs & Hat Special!

That's 144 issues. Complete with supporting code and media files.



Free Shipping!

Only \$249.95 or \$24.95 each.

POCKET REF
Thomas J. Glover

The Nuts & Volts Pocket Ref

All the info you need at your fingertips!

This great little book is a concise all-purpose reference featuring hundreds of tables, maps, formulas, constants & conversions. AND it still fits in your shirt pocket!

Only \$12.95

Visit <http://store.nutsvolts.com> or call (800) 783-4624

Beyond the Arduino

Power: Less is More

This month, we focus on common ways to reduce the power consumption of our projects so they can run for longer on less. That's right! We'll be putting our microcontrollers to sleep. No lullaby needed ...

Keep It Down

One of the things that I really like about building embedded systems is that they are portable. I can pack a host of functionality into a small microcontroller-based project, pop it into a weatherproof container, and deploy it pretty much wherever I want it. The projects don't need to be tethered to a power point as their consumption is relatively low. Attach a small battery pack and they'll run for days. All you need to do is switch the battery pack once a week, and you have a truly remote system.

How does that sound? Well, not ideal in my world, actually. I'm not a fan of clambering around changing the batteries in all my projects every weekend. Therefore, we're going to look at ways to reduce the power consumption of our embedded projects, so that a project can run for months without needing a battery change.

There are a number of ways to reduce the power consumption of an embedded project — each with different levels of complexity and, of course, degrees of effectiveness. Here are some that we **won't** look at in detail in this article. They seem like common sense, but I see project after project where they are overlooked (including some of my own):

- **Component Choice:** Different components use differing amounts of current. How often do you read the datasheet of an LED to see the current draw, and choose the LED on that basis?

- **Power Conditioning:** Implement more efficient

By Andrew Retallack
www.crash-bang.com

8

power conditioning by using a low dropout voltage regulator instead of a linear regulator. Also make sure you aren't stepping voltage down by more than a couple of volts because this results in wasted energy and current consumption.

- **Run Off Battery:** Establish if you need to control your voltage. I've run many simple projects straight off the battery, eliminating all the leakage from the power conditioning circuitry. This won't work if you need a fixed reference voltage, for example, in ADC (analog-to-digital converter) applications — unless you include a circuit to provide a reference voltage.

- **Turn It Off:** Turn off the components that you don't need. Do you really need a power indicator LED on your project? If you're logging temperature every hour, does your temperature sensor need to be powered all the time, or can you turn it on just before taking a reading?

These are relatively simple steps, and can be retrofitted to an existing project to an extent. The rest of this article will look at measures you need to consider when you're still in the design phases of your project — unless you really enjoy re-engineering your completed projects! They are fundamental considerations; they also have the biggest impact on your project's consumption.

Choose Your Volts

The ATmega328P is a 5V microcontroller, right? The Arduino runs at 5V, our projects so far have run at 5V, so it must be! Wrong! The ATmega328P can run at anything from 1.8V to 5.5V. We chose 5V as it's the most common voltage for the microcontroller to run at. Many of the more popular components and modules we use need a 5V supply, so it makes sense to run at 5V — but not if you want to reduce your consumption and run off a battery for months.

Look at the extract of the MCP79400 datasheet in **Figure 1** (this is the real time clock module we used in the September 2015 article). The graph shows the current consumption at voltages of 1.3V, 3.0V, and 5.0V. Look how that figure climbs as the voltage increases! Sure, we're talking nanoamps, but take note of the magnitude

of the increase between 3V and 5V — nearly nine times the consumption.

A proportional increase in current as voltage increases shouldn't really be a surprise. If you go back to principles and consider Ohm's Law, it makes sense. Ohm's Law states that $I = V/R$. If we assume a constant resistance (R), then as the voltage (V) increases, so will the current (I). Of course, there must be other factors at play within a complicated integrated circuit that contribute to this increase, but at a very simplistic level it's something that we should expect.

How Low Should I Go?

So, to reduce current consumption, we need to lower the voltage. However, you usually can't simply drop the voltage to the microcontroller's minimum. There are a few things to consider when choosing the voltage that you want to run your project at. In embedded systems, there are always trade-offs and/or complications, and this is no different. Here are three things to consider:

Other Modules and Components: In many projects, you'll need to connect the microcontroller to other modules and components, and so need to consider their limitations. The LM35 temperature sensor, for example, only operates between 4V and 20V. So, it wouldn't be much good in a low power project. If you want to drop below 4V (3.3V is a common voltage), then you'll need to find another part. In the past, I've used the LM60 from Texas Instruments which operates from 2.7V. Of course, as soon as you start switching parts out, you risk heading down the rabbit-hole of price vs. precision vs. operating voltage vs. features ... you get the picture!

So peg the non-negotiable parameters down and work around those. In the September 2015 article, you may remember that one of the reasons I choose the MCP79400 as a real time clock over the more popular Maxim DS1307 is that the DS1307 only operates at 5V, whereas the MCP79400 operates over a range of 1.8V to 5.5V. I can therefore use it in low power applications.

Battery Performance: You probably know that a battery's voltage rating is "nominal." Commonly, it will start out higher than the rated voltage and will then drop below the rated level. **Figure 2** shows the voltage over time of a standard Duracell Coppertop AA 1.5V battery under a constant current load. Under a 5 mA load, nearly half the battery's life is spent below the 1.3V level.

You need to plan for this voltage degradation, and ensure that your microcontroller and accompanying components are still able to perform at the voltages that

MCP79400/MCP79401/MCP79402

TABLE 1-1: DC CHARACTERISTICS

DC CHARACTERISTICS (Continued)			Electrical Characteristics: Industrial (I): $V_{CC} = +1.8V$ to $5.5V$ $T_A = -40^\circ C$ to $+85^\circ C$				
Param. No.	Symbol	Characteristic	Min.	Typ. ⁽²⁾	Max.	Units	Conditions
D15	IBATT	Timekeeping Backup Current	—	—	850	nA	$VBAT = 1.3V$, $V_{CC} = V_{SS}$ (Note 1)
			—	925	1200	nA	$VBAT = 3.0V$, $V_{CC} = V_{SS}$ (Note 1)
			—	—	9000	nA	$VBAT = 5.5V$, $V_{CC} = V_{SS}$ (Note 1)
D16	IBATDAT	VBAT Data-retention Current (oscillator off)	—	—	750	nA	$VBAT = 3.6V$, $V_{CC} = V_{SS}$

Figure 1: Extract from MCP79400 datasheet: power consumption vs. voltage.

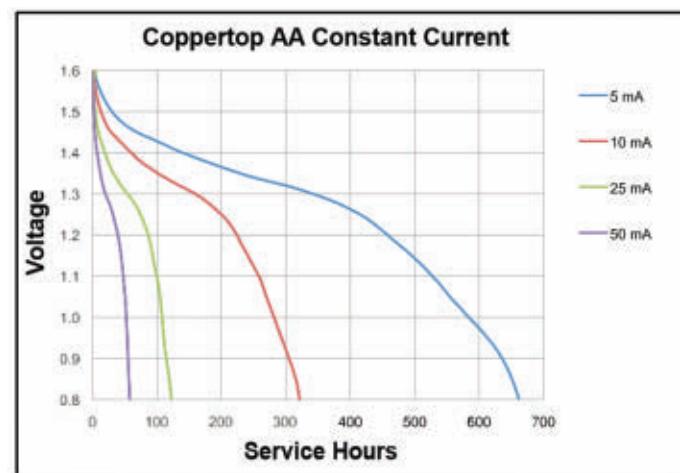


Figure 2: Extract from Duracell AA datasheet: voltage over time under load.

Fuse Bytes

AVR microcontrollers have certain settings that aren't accessible through code, but need to be configured using a programmer. Atmel calls these "fuse bytes" or simply "fuses" — a name that has caused much confusion.

Fuse bytes control a range of settings that you're not likely to use in your projects — many are related to how the MCU is booted, reset, or programmed — as well as a set of useful ones, for example, that control the BOD trigger level and specify the clock source and its speed.

If you have a programmer that integrates with Atmel Studio, you can take a look at the fuses in the Device Programming dialog (from the Tools menu). If your programmer doesn't natively work with Atmel Studio, then you'll need to use an external program like AVRDUDE (which you're likely already using to program your microcontroller). It's beyond the scope of this article to dive into detail, but there is a fair amount of online resources available. Take a look at the Resources box for a detailed online article that I wrote.

Figure 3: Sleep mode characteristics.

Sleep Mode	Which Clocks are Active?					Which Oscillators are Active?		What Interrupts can Wake the MCU?					
	Asyc Timer	ADC	I/O	Flash	CPU	Timer	Main	INT1/INT0 Pin Change	I ² C Address Match	WDT	Timer 2	ADC	SPI/M EEPROM Ready
Idle	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
ADC Noise Reduction	✓	✓				✓	✓	✓	✓	✓	✓	✓	✓
Extended Standby	✓					✓	✓	✓	✓	✓	✓		
Power-Save	✓					✓		✓	✓	✓	✓		
Standby							✓	✓	✓	✓			
Power-Down								✓	✓	✓			

are likely towards the end of the battery life. You could work around this with a buck-boost converter to ensure a constant voltage output but, of course, this comes at a cost: The converter and its components constantly consume power.

Brown-Out Detection (BOD): This functionality allows you to specify a minimum voltage at which your microcontroller should operate. If the voltage drops below that level, then the system enters a “reset” state and won’t operate until the voltage increases again. I don’t often use this, but it could be an important consideration if your project is one that will misbehave at low voltages.

For example, some of your components may stop working below a certain voltage but not others, causing the project to behave in an unpredictable (even dangerous) manner. You can configure the BOD level by setting specific fuse bits (see the **sidebar** on Fuses).

We’ve touched on voltage considerations, and understand that it can significantly impact the power consumption of a project. Our project this month will get us into a little more detail, but for now, let’s move on to the design consideration that will give us the greatest current consumption savings.

Rock-a-Bye AVR

Okay, I admit, that was a really bad heading to take us into the details of the most important power-saving design methodology: the use of sleep modes. We’ve been building up to this discussion over the past few articles as we covered topics such as interrupts and timers, and we’re going to need all that we covered in those articles to tackle sleep modes.

Let’s start by understanding what it means when a microcontroller “sleeps.” You could draw an analogy with animals that go into hibernation: Their systems shut down or slow down to the minimum needed to keep them alive, and as a result they consume very little energy and need very little (if anything) to eat. When a microcontroller is put into sleep mode, it too shuts down certain of its modules and peripherals, resulting in drastically reduced

power utilization. Just as the onset of spring triggers the emergence of animals from hibernation, an interrupt triggers the end of an MCU’s digital sleep. When designing an embedded system, it’s common to spend most of the time in sleep mode, only waking up when something needs to be done; for example, a temperature reading being taken.

Most microcontrollers have a range of sleep modes, but they do vary by specific microcontroller. What we’ll cover in this article has some generic applications, but is ultimately specific to our trusty ATmega328P. What systems does the MCU shut down when it goes to sleep, and if it is asleep what kind of interrupts are able to wake it up again? I’m glad you asked, because to understand that we need to get into some technical detail. We’re going to spend some time looking at the various sleep modes. These can seem confusing at first, so if you enjoy a strong cup of coffee like I do, now would be a good time!

Types of Sleep Mode

The ATmega328P has six types of sleep modes — each with different combinations of active modules and wake-up sources. When designing a project, you’ll need to spend some time deciding which sleep mode is most appropriate for you. I’ve often found that the one I wanted to use (due to the power consumption) wasn’t suited to my project, as I wasn’t able to wake it up with the type of interrupt I needed to. I’d love to be able to give you a definitive way to decide on the best mode to use, but the reality is that it is completely project dependent.

Before you spend time on the details of the various modes below, consider what it is that drives the consumption of power in a microcontroller sub-system or module. If you’ve been really good at reading between the very blurry lines in this series so far, you may have guessed that it’s the clock. Without a clock source, the microcontroller and its modules will not function.

We’ve discussed that tick-tocking drives the CPU, the ADC, and the timers; how the clock is critical to timing on the UART; and the I²C interface. Without a clock pulse to “clock” instructions in and out of the CPU, nothing will happen. Take a look at the **sidebar** “How Many Clocks?”

to get a little more insight into the clocks that operate inside your ATmega328P. They are what we use to put the microcontroller to sleep. As you continue reading, refer to **Figure 3** for a summary of the sleep modes.

Idle Mode

This mode results in the least savings because it only shuts down the CPU and the Flash memory interface, leaving everything else running. It does this by disabling the clk_{CPU} (CPU clock) and the $\text{clk}_{\text{FLASH}}$ (Flash clock). This means that the MCU can be woken by pretty much any internal or external interrupt — including ADC-driven, timer-driven, and USART-driven interrupts. From this perspective, it is the most flexible of the six modes.

ADC Noise Reduction Mode

This mode is designed to improve the accuracy of ADC readings, and is not really a power-saving method. By shutting down the CPU (clk_{CPU}), the Flash interface ($\text{clk}_{\text{FLASH}}$) and the I/O module (clk_{IO}) on-chip noise is reduced, resulting in higher resolution ADC measurements. If your ADC is enabled, entering this sleep state will automatically start an ADC read operation — and then the ADC Conversion Complete interrupt will wake the microcontroller up again. Other interrupts are able to wake the MCU, but typically wouldn't be needed.

Extended Standby Mode

Before continuing, I need to comment that I have a great deal of sympathy for the engineer that was tasked with naming these various modes. It is impossible to name them in a way that is at all descriptive of their function. So, as you read on, wonder at their cryptic naming and be grateful that neither of us had to choose the names!

Extended Standby mode is a good middle-ground mode. It is efficient, but retains some of the flexibility that I find useful. All the clocks except for the Asynchronous Timer clock (clk_{ASY}) are stopped, meaning that if Timer2 is running, it will continue to run during sleep. This allows you to use Timer2 to wake the microcontroller up. The main oscillator also remains running, so the MCU will wake up quickly — in six clock cycles.

Power-Save Mode

This is more efficient than Extended Standby mode, as the main oscillator does not run. Otherwise, it is identical to the Extended Standby mode. The impact of the main oscillator not running is that the MCU takes longer to wake up — pretty much the same time as it does after a power-on.

Standby Mode

Standby mode is the second most efficient mode, and also the least flexible. All clocks are stopped (clk_{CPU} ,

SMCR – Sleep Mode Control Register

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	SM2	SM1	SMD	SE
								Sleep Mode
								Sleep Enable

MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0
Function	-	BODS	BODSE	PUD	-	-	IVSEL	IVCE
								Brown-Out Detector Sleep

PRR – Power Reduction Register

Bit	7	6	5	4	3	2	1	0
Function	PRTWI	PRTIM2	PRTIMO	-	PRTIM1	PRSPI	PRUSART0	PRADC
								TWI (I ² C)
								Timers
								SPI
								UART
								ADC

Figure 4: Summary of sleep and power registers.

How Many Clocks?

Your microcontroller has five clocks — that's right, five: the CPU clock, the Flash clock, the I/O clock, the Asynchronous Timer clock, and the ADC clock. It took me a long time to get my head around all of these clocks and how they worked. If you feel similar, then take heart! You've been working with most of these already! In essence, the ATmega328P has a Clock Control Unit that takes the clock source (an oscillator) and distributes it to each of these clocks.

Here's a quick round-up of the clocks:

- **CPU Clock (clk_{CPU}):** This drives the core of the microcontroller, along with all the core system registers, stacks, and so on.
- **I/O Clock (clk_{IO}):** This is used by I/O modules — external interrupts, timers, and serial modules (SPI/USART0).
- **Flash Clock ($\text{clk}_{\text{FLASH}}$):** This controls the Flash memory interface's operation.
- **Asynchronous Timer Clock (clk_{ASY}):** This allows Timer2 to work in asynchronous mode; in other words, it is not linked to the same clock source as the other clocks, but runs off an external source. This is a topic we'll cover at a later stage.
- **ADC Clock (clk_{ADC}):** This clock controls the operation of the ADC (analog-to-digital converter), allowing it to run in ADC Noise Reduction mode when other clocks are disabled.

Clocks and Oscillators

Something that I struggled with initially was the difference between a clock and an oscillator. An oscillator is a clock source — think of it as your car's engine. The oscillator can drive multiple clocks; a bit like when you engage gear on your car, the engine drives multiple wheels. Oscillators consume power (small amounts), so you'll see that some sleep modes allow you to disable them. The disadvantage to disabling them is that it takes longer to wake from sleep (a bit like starting your car engine then engaging gear, as opposed to simply engaging gear).

SM2	SM1	SM0	Sleep Mode
0	0	0	Idle Mode
0	0	1	ADC Noise Reduction Mode
0	1	0	Power-down Mode
0	1	1	Power-save Mode
1	1	0	Standby Mode
1	1	1	Extended Standby Mode

Table 1: Sleep Mode Settings.

$\text{clk}_{\text{FLASH}}$, clk_{IO} , clk_{ADC} , clk_{ASY}) and only an External interrupt, an I²C address-match, and a watchdog timer interrupt can wake the microcontroller. Other than the watchdog timer, the microcontroller cannot wake itself using an internal interrupt source. It needs an external stimulus. The main oscillator is kept running, so the MCU wakes in only six clock cycles.

Power-Down Mode

Power-down mode is the most efficient mode, and is almost identical to Standby mode – the only difference being that the main oscillator is also shut down. This means that the wake-up time is of a power-on type of duration.

Putting Your Microcontroller to Sleep

Once you've overcome the challenge of understanding and choosing a sleep mode, it's going to seem ridiculously easy to put your microcontroller to sleep. For a start, there's only one register that you need to configure: the aptly-named "Sleep Mode Control Register" (SMCR). Secondly, there are only three steps that you need to follow to enter sleep mode. Let's start out by looking at the registers in **Figure 4**.

Sleep Mode Control Register

The Sleep Mode Control Register simply allows you to

```
int main(void)
{
    //Initialise the IO Pins
    LED_PORT_DIR |= (1<<LED_BLINK_PIN) | (1<<LED_SLEEP_PIN);
    //Set pin to Output

    //Initialise the Timer2
    Timer_Init();

    //Initialise Sleep Mode
    set_sleep_mode(SLEEP_MODE_PWR_SAVE);
    //Set Sleep Mode to "Power-Save"

    sei();    //Enable Interrupts

    while(1)
    {

```

configure which of the sleep modes you want your microcontroller to enter. **Table 1** summarizes the settings.

Once you've set the sleep mode using the **SM2-SM0** bits, you then need to write a "1" to the **SE** bit. This enables sleep mode, but doesn't activate it. This requirement to enable sleep mode serves as a safety net to prevent you from sending your MCU to sleep accidentally.

Turning Out the Lights

It takes no time at all to turn out the lights and put your MCU to sleep. That is, once you've planned how you're going to wake the microcontroller up again! This code excerpt shows the required steps:

```
set_sleep_mode(SLEEP_MODE_PWR_SAVE);
//Set Sleep Mode to "Power-save"
sei();
//Enable Global Interrupts
sleep_mode();
//Enter Sleep Mode
```

I'm sure that the above doesn't look like what you expected. For a start, no registers are being set. Well, that's probably because I left out an important line:

```
#include <avr/sleep.h>
```

That's right, there's a useful *include* file that does all the heavy lifting for us. So, in fact, we didn't *really* need to know the details of the Sleep Mode Control register, but I do think it's useful to understand what's happening behind the scenes.

Let's dissect the code above:

`set_sleep_mode(SLEEP_MODE_PWR_SAVE);` This sets the sleep mode, and is the same as if we used the registers directly: `SMCR = (1<<SM1)|(1<<SM0);`

`sei();` You're already comfortable with interrupts, so

```
LED_PORT ^= (1<<LED_SLEEP_PIN);
//Toggle the LED_SLEEP

sleep_mode();    //Go to sleep

/**Execution continues here once the MCU
//has woken, after the ISR has run.

//If a reasonable delay has elapsed, then
//toggle the LED_BLINK
if (delayCounter >= 10)
{
    LED_PORT ^= (1<<LED_BLINK_PIN);
    //Toggle LED
    delayCounter = 0;
    //Reset delayCounter
}

}
```

Listing 1: Project 1: The `main()` function.

this command will look familiar. Remember that this just enables *global* interrupts. You'll need to have already configured your interrupt source (e.g., the ADC/external interrupt/timer interrupt).

`sleep_mode();` This final function (well, macro actually) really does three things:

- Sets the Sleep Enable bit: $SMCR |= (1 << SE)$;
- Puts the microcontroller to sleep: `_sleep()`;
- Clears the Sleep Enable bit; this will happen when the MCU wakes up: $SMCR \&= \sim(1 << SE)$;

That's it! We've successfully put the microcontroller to sleep. So, what happens when it wakes up?

When the Microcontroller Wakes Up

We've configured one (or more) interrupts to wake the MCU from its sleep mode. Now what? When the interrupt triggers, the microcontroller wakes up (taking at least 10 clock cycles to get up to full steam) and enters the Interrupt Service Routine (ISR). When the code in the ISR has completed, execution returns to the statement immediately after the sleep instruction that initially put the microcontroller to sleep. As we usually try to keep code in the ISR as short as possible, it's likely that we'd want to use the ISR to set a flag that then gets processed back in the main `while(1)` loop.

Once we've finished executing the code that we woke up to run, we simply put the microcontroller back to sleep. I usually put the sleep commands in the main `while(1)` loop so the MCU goes back to sleep on each iteration of the loop.

Yep. That is all there is to sleep modes. I'm sure you're chomping at the bit to see this all in action, so let's dive into our first project.

The First Project: Sleep

I've lined up two projects this month. The first is a simple one that visually shows us the workings of sleep mode. The second one I'm going to leave to you to play with – with a bit of guidance, of course. Let's get cracking!

The first project has two LEDs attached via current-limiting resistors to **PB0** and **PB1**. **Listing 1** shows the bulk of the code and **Figures 5** and **6** show the layout. The first LED (I call it *LED_BLINK* in the code) is blinked by an interrupt on Timer2. Last month, we worked with Timer1, but Timer1 isn't able to wake a microcontroller from sleep. So, we need to use Timer2 for this project. We use it in the same way, although the control registers are slightly different.

Also, it is an eight-bit timer, so the most we can count up to is 255. This means that the LED will flash pretty quickly! To combat that, I've used the Timer's ISR to increment a variable (*delayCounter*) on each interrupt:

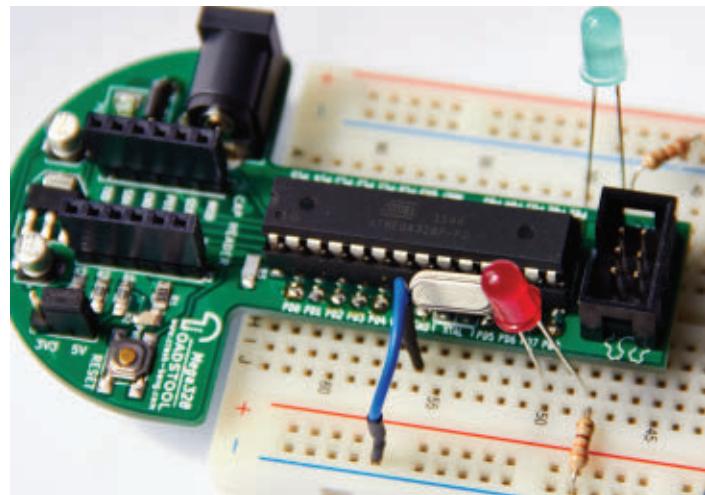


Figure 6: Project 1 using a Toadstool.

when `delayCounter` reaches 10, I toggle the LED. You could also use a slower crystal. If you're currently using a 16 MHz one, the lowest you can go without reconfiguring the fuse bytes is an 8 MHz crystal.

The second LED (*LED_SLEEP*) is there to prove to you that the microcontroller is actually going into sleep mode. I needed it to prove the same to me! This LED is toggled before the sleep command is executed. If the sleep command succeeds, then execution of the program will halt until the timer interrupt wakes it; this delay causes the LED to blink (very fast). If the microcontroller doesn't go to sleep, then execution will continue and the LED will be toggled a few lines of code later, causing the LED to glow constantly (although dimly). Test this functionality by commenting out the *sleep_mode()* command and watching how the LED behaves.

The rest of the code is pretty straightforward and should consolidate what we've discussed earlier in this article. The problem with this project is that it doesn't really demonstrate how the sleep mode is reducing the power consumption of the project. It's for that reason we

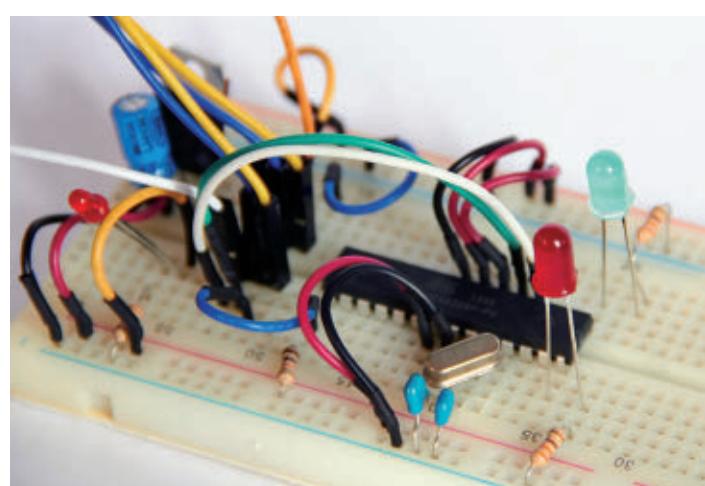


Figure 5: Project 1 on a breadboard.

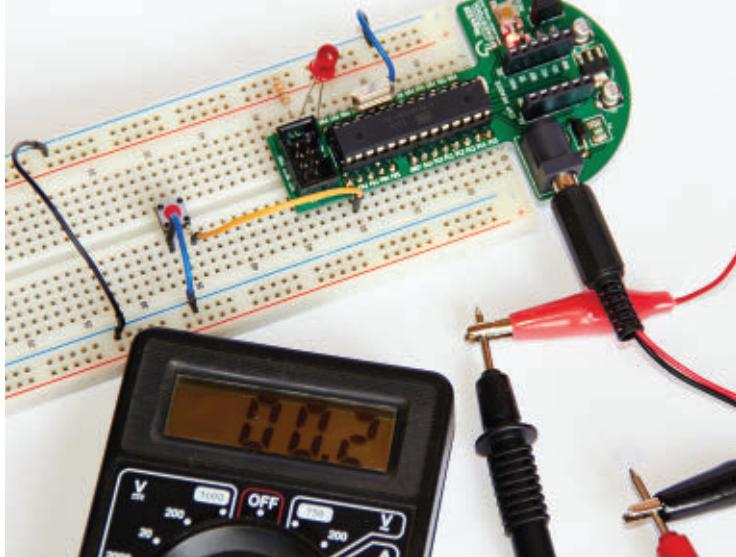


Figure 7: Measuring current draw in Project 2.

have a second project!

Squeezing the Last Drops

We've covered the MCU modules that are disabled during sleep mode, but what about modules you don't need while the microcontroller is running? Additionally, what if the sleep mode you're using still has modules in an "awake" state when you don't actually need them? All sleep modes can be woken by an I²C (TWI) address match, for example, which means a part of the TWI module is running even when asleep. If you don't use TWI at all, then surely it makes sense to disable it at the outset. Going to this level is really squeezing the last drops of efficiency from your project, but over a period of time it all adds up. There are two registers that enable us to disable unused modules: the MCU Control Register (MCUCR) and the Power Reduction Register (PRR); refer back to **Figure 4**.

MCU Control Register

The MCU Control Register — for our power-saving purposes — gives us the ability to enable or disable the BOD while the microcontroller is asleep. The BOD continuously monitors the voltage supply to the MCU, but if that's not needed during sleep, then it makes complete sense to turn it off. The **BODS** and **BODSE** bits are used to control this, in a somewhat convoluted manner. To disable BOD during sleep, you need to:

1. Set both **BOD** and **BODSE** to "1."
2. Set **BOD** to "1" and **BODSE** to "0" (the latter within four clock cycles).
3. Enter sleep mode within three clock cycles; in other words, almost immediately!

Alternatively, you can use the macro that's included in

Resources

Understanding AVR Fuse Bytes:
www.crash-bang.com/programme-avr-fuse

Atmel Video — Power-Saving Techniques:
www.crash-bang.com/Atmel-Low-Power-Video

Nuts & Volts Forum:
<http://forum.nutsvolts.com>

the "sleep.h" library to handle this complexity for you:

```
sleep_bod_disable();
```

Power Reduction Register

The PRR allows you to disable the TWI (I²C) module, Timers 0, 1, and 2, the SPI module, the USART0, and finally the ADC. Each of these are disabled by writing a 1 to the corresponding bit in the register, and enabled by writing a 0. The serial modules (I²C, USART0, and SPI) need to be re-initialized if they are disabled and then re-enabled. Also note that if you're debugging your project with a debugger (using the debugWire interface), then you should not shut the SPI module down. It's needed to handle the debugging interface.

The PRR doesn't only affect sleep modes. It's a good way to keep consumption down when the microcontroller is awake, as well. Now that we've touched on these in theory, let's put them into practice!

The Second Project: Low Power

This simple project uses a pin change interrupt to toggle an LED. When you press the momentary pushbutton connected to **PB1**, the LED on **PB0** lights up. You're probably thinking that sounds pretty boring; after all, we're now eight articles into the series. Well, this is more of an experiment than a useful project. The idea is that you'll measure the impact of the various power-saving techniques we've discussed so far. Hopefully, things are starting to sound a little more interesting!

First, connect the LED to **PB0** (through a resistor as usual) and a pushbutton to **PB1** (connect the button's terminal to GND, as we're using the ATmega's internal pull-up resistor). Then, connect a digital multimeter to the project, so that it is in line with the VCC power supply (i.e., the power needs to "flow through" the DMM). Switch the DMM to measure amps and adjust the scale as you need. We're ready to start our experiments now.

After each of the below, record the readings on the DMM (**Figure 7**). The experiments are clearly commented in the code, so you should find them easily enough.

Experiment 0: Obtain a baseline reading. We haven't implemented any power savings yet, so take a baseline reading to see just how effective the savings are.

Experiment 1: Uncomment the line of code that sets

the PRR register, then compile and upload. Here, we're disabling all the modules and peripherals that we don't need to use in order to save power. Take a reading.

Experiment 2: Uncomment the line of code that activates sleep mode, then compile and upload. Here, we're doing what we did in the first project and putting the microcontroller to sleep between interrupts. The only difference is that the interrupts are manually driven by a button, not a timer. Take another reading.

Experiment 3: Uncomment the line of code that disables the BOD sensor, then compile and upload. Now, the Brown Out Detector will be disabled during sleep mode as well. Take a reading.

Experiment 4: The final experiment is to lower the voltage into the project. For this, you'll need a 3.3V voltage regulator (if you're working on a breadboard), or if you're using a Toadstool, you'll need to switch the jumper from 5V to 3V3 (with the power disconnected, of course). Take your final reading.

Final Step: Please let us all know how low you got your power! Pop on over to the N&V forum, leave a comment on the website, or drop me a line. I'd love to hear your results.

What We Didn't Cover and What's Next

There just isn't enough space here to cover (in detail) all the ways to reduce power consumption, but we've covered the ones with the greatest impact. If you'd like to research more, then take a look through the datasheet as well as online. Look for discussions on pull-ups on pins that aren't connected, or how changing your clock frequency plays a role. There's a great deal of information out there!

We've come a long way over the last several months, and I've had some really encouraging feedback

A complete kit to go with this article can be purchased online from the **Nuts & Volts** Webstore @ www.nutsvolts.com or call our order desk at 800-783-4624.

and helpful suggestions on topics to cover. Please keep the thoughts and the suggestions coming! There are still a bewildering number of areas to dive into. Let me know what **you** would like to read about. I'm planning to dig into the SPI protocol (I'm playing with an SPI 3D accelerometer at the moment), as well as some debugging techniques, other AVR microcontrollers, using LCD displays, and a whole host more over the coming months.

Thanks for joining me this month, and keep on experimenting! **NV**



3-POSITION 10 AMP ROTARY SWITCH

1-pole, 3-position. 6mm dia. shaft. 9mm threaded bushing. 10 Amp/ 125 VAC, 6 Amp/ 250V T85. Plastic body, shaft and bushing.

CAT# RSW-310

\$1.70
each

GREAT PRICE! 2 GB MICRO-SD MEMORY

Kingston SD-CO2G 35.

CAT# MSD-2GB

\$2.95
each



10 for \$2.80 each
100 for \$2.50 each

12VDC 15A POWER SUPPLY

Protek Power

PUP22012.

Table-top

style switching supply.

4-pin plug that mates with Kycon P/N KPJX-4S-S or equivalent. Overvoltage, overcurrent protection. Detachable 3-conductor power cord. CE, cULus.

CAT# PS-12152

\$39.75
each

16 for \$37.00 each

PLEXIGLASS FRAME

Attractive, unique, free standing acrylic photo frame made of two crystal clear acrylic blocks held together by neodymium magnets. Great for displaying baseball cards, business cards or a photo cut to fit. Trapezoid shaped, it can stand on any of its four edges. 5.95" high x 5.93" wide x 0.95" deep.

CAT# PF-2



\$5.95
each

10 for \$5.50 each

0-15 VDC PANEL METER

Shurite 350 Series# 3108.

1.5 inch edgewise panel meter. Polystyrene case with metal scale, white field, printed black. Age and moisture resistant. Cutout size: 0.531" x 1.656". Includes black bezel & mounting hardware.

CAT# PM-115

\$15.50
each



MINI-GRABBER TEST LEADS, 24"

Fully-insulated, spring-loaded 2" mini-grabbers for connection to small terminals in tight places. Flexible 24" leads w/ gold-flashed banana plugs. Set of red and black leads.

CAT# TL-9



\$5.95
each

SPECIAL! LIGHTNING CHARGE / SYNC CABLE, 1M

1 meter (3.28') cable to charge and sync newer Apple iPhones / iPads / iPods. 8-pin lightning plug to USB plug.

CAT# IPD-41

\$4.75

each



22 GAUGE RED/BLACK ZIP CORD

Automotive grade "GPT" zip cord with red and black conductors. 80°C, flame-retardant PVC insulation. 10 ft minimum order.

CAT# WRB-22

50' for 13¢ per ft.

100' for 12¢ per ft.

500' for 11¢ per ft.

10' for **\$1.40**

READER FEEDBACK

Continued from page 7

"breathes" while oscillating back and forth.

If any coils of the hairspring touch each other, it effectively shortens the length of the spiral and radically increases the spring rate and natural frequency of the oscillator. Even small spurious magnetic fields will cause the coils to stick together when magnetized and make huge increases in the rate of a watch.

It is not easy to demagnetize the hairspring in an assembled watch, but it can be done by immersing it in a demagnetizer several times. Look at the hairspring of a magnetized watch while running and you will see the coils sticking to each other and destroying its spiral profile.

So, the answer to the first question is that the oscillator is isochronous and therefore immune (within limits) from changes in the driving force resulting from increased efficiency or any other reason, and is only influenced by the ratio of the spring rate to the moment of inertia of the balance wheel. A crystal controlled oscillator is somewhat

analogous electrically.

ANSWER #2. EDDY CURRENTS ARE NOT IMPARTING ENERGY TO THE MAINSPRING.

Eddy currents cannot impart energy to the mainspring, and even if they could, would not affect the rate of the watch. Eddy currents act as a mechanical brake and require relative movement between a magnetic field and a conductive – but not necessarily magnetic – material.

If the balance wheel assembly were highly magnetized, it could generate a little eddy current braking, resisting the oscillatory movement, but isochronism would keep the rate constant. You can make huge changes to the mainspring of a good watch and leave the rate relatively constant.

ANSWER #3. MAGNETIZED MOTORS DO NOT RUN MORE EFFICIENTLY.

Magnetized motors may run less efficiently because of the spurious braking action of the unwanted eddy

currents. Note that motors that use more non-conductive material like plastic in their structure can improve efficiency by minimizing eddy current braking.

I was not as brief as I should have been, but I wanted to emphasize how hidden magnetic fields can degrade mechanical devices.

David Miller
Boulder City, NV

You've really considered the questions posed in my editorial, David. You apparently know your way around a mechanical watch. It is much appreciated.

As an aside, I've been reviewing the various electro-mechanical engines developed by Seiko over the past few decades, including their spring drive, which is part mechanical, part quartz.

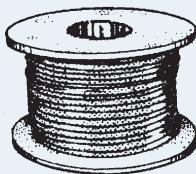
Bryan Bergeron

Spiraling Together

I love Bryan Bergeron's column. It's the first thing I read.

CLASSIFIEDS

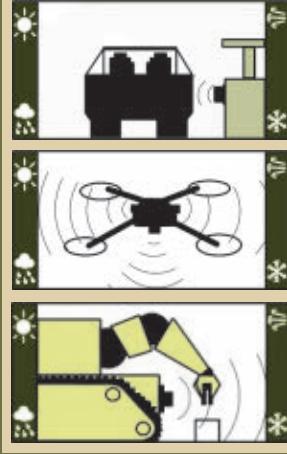
WIRE/CABLE



ANAHEIM WIRE, INC.
Master distributor of electrical and electronic wire and cable since 1973. Items available from stock: Hook up wire, Shrink tubing, Cable ties, Connectors, etc. Wire cut & strip to specs, twisting, striping. If interested, please call **1-800-626-7540**, FAX: 714-563-8309. Visa/MC/Amex. See us on the Internet: www.anaheimwire.com or email: info@anaheimwire.com.

ROBOTICS

Need sensors?



www.maxbotix.com

LIGHTING



LED lighting for displays, props, scenery, signs, models and more.

www.j2ledlighting.com



Subscribe today!

Like what you see? Go to www.nutsvolts.com

HARDWARE WANTED

DEC EQUIPMENT WANTED!!!

Digital Equipment Corp. and compatibles.
Buy - Sell - Trade

CALL KEYWAYS 937-847-2300
or email buyer@keyways.com

In regards to the Nov 2015 editorial, the hairspring that makes the balance wheel rotate back and forth to provide the timing pulses is an Archimedes spiral with constant distance between each turn. Usually when this spring is magnetized, the watch completely stops.

In the case of his watch, I suspect that the additional field provided by the scissors was enough to cause two adjacent turns to stick together, thereby making the effective length of the spring to be shorter and the balance wheel to oscillate faster. This could be due to a partially magnetized spring or one with slight mechanical defects. If you run the watch through a demagnetizer, it should eliminate the first of these two.

Another demagnetizer I've used is re-purposed degaussing coils from old color TV sets. They consist of a coil of wire around the picture tube in series with a positive temperature coefficient thermistor across the 120 VAC line. As the thermistor heats up, its resistance increases, damping out the 60 Hz magnetic field much like a capacitor discharge unit.

The 20 (?) inch diameter coil can be twisted into smaller coils, making more turns and thus stronger demagnetizing fields.

William Ridgway AH6TW

Thanks for taking the time to solve my watch mystery. Makes sense. The demagnetizer did the trick – for both the watch and the scissors.

And thanks for the excellent idea of using the degaussing coil. I remember the coil kicking in with a pop on the old color CRTs. The problem today is finding a CRT. Even my parents have a flat screen these days.

Bryan Bergeron

Magnetic Personality

Referring to the Nov 15 Developing Perspectives column, my

hypothesis regarding the speed increase of a magnetized watch is that when the spring is magnetized, there is an additional force that tends to keep the spring more closely wound up.

This has the effect of more stiffness, which increases the spring

constant and therefore increases the frequency.

Gerard te Meerman PhD
Netherlands

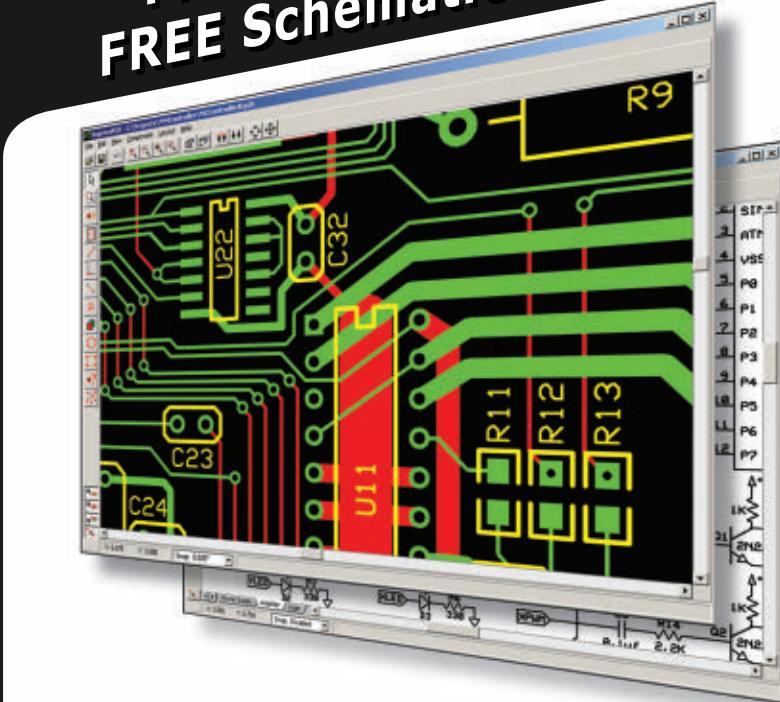
*I appreciate your hypothesis.
Makes sense to me.*

Bryan Bergeron

\$51 For 3 PCBs

FREE Layout Software!

FREE Schematic Software!



01 DOWNLOAD our free CAD software

02 DESIGN your two or four layer PC board

03 SEND us your design with just a click

04 RECEIVE top quality boards in just days

expresspcb.com

>>> QUESTIONS

Clipping The Spike

I'm using an Arduino to control a set of relays, and both the microcontroller and relays share a 9V power bus. I plan to put a diode across the relays to clip any reverse voltage spikes. Is there anything else I should consider to prevent a spike from the relays from shutting down the processor?

#1161 **Ronald Miller**
Kansas City, MO

Capacitor Forming

I'm trying to resurrect an old Halicrafter's communications receiver from at least the '50s. I'm planning to replace the electrolytic capacitors in the power supply with capacitors I salvaged from a more recent TV set.

However, I've read that electrolytic capacitors — once formed at a certain voltage — can take months, if not years, to reform at a new voltage. Until then, the capacitor value can be significantly off from what's on the label.

Can anyone shed some light on this, and any thoughts on whether I'll risk damaging the receiver if I use the caps formed at the higher voltages found in the TV circuit?

#1162 **Nicolas Berger**
Birmingham, AL

Switching To A Switching PS

I found a fantastic tube amp for my stereo system at a garage sale, but later discovered that the power transformer is shot. Can I use an inexpensive switching power supply

to provide the high voltage, without distorting the output? I'd rather not spend \$100+ on an old fashioned boat anchor transformer if I can avoid it.

What are the pros and cons? Any advice would be appreciated.

#1163 **John Ullrich**
Santa Fe Springs, CA

>>> ANSWERS

[#8152 - August 2015] Dim Bulb

What's the difference circuit-wise between a "dimmable" and a "non-dimmable" 110V home LED light bulb?

The non-dimmable variety use a largish electrolytic capacitor in the power supply to generate a DC voltage, and then the LEDs are driven with a constant current circuit. So, because of the constant current, the LED output is unaffected by supply voltage, so the brightness won't change with voltage (or dimming). Dimmable LEDs only have a small film capacitor; the control circuit sets the current according to the input voltage (so they pulsate at mains frequency) and when dimmed the light output is chopped up similar to an incandescent bulb. The above applies to screw-in bulbs; larger LED supplies may still use electrolytic capacitors, but within these another circuit measures the incoming duty cycle and adjusts the LED current to match.

Bob Turner
Salamander Bay, Australia

[#9152 - September 2015] Is It Magic?

I have satellite radio (Sirius) aux audio output connected to DLO TransPod aux input. The TransPod is normally used for iPod to FM radio in a car. I am using it in my house.

The audio is transmitted on a frequency to my FM radio (50 ft away) with no power to the TransPod! How is this possible?

It appears the output of the Sirius unit is supplying "phantom power" to the TransPod. IOW, there is a small DC voltage (1-2 VDC) present on the output signal that the TransPod is using as its power source. In a way, it is "magic" as phantom power schemes are regularly used in audio PA systems to supply power to electret microphones without using external batteries.

Ken Simmons
Auburn, WA

[#9153 - September 2015] Geocache Container With Flashing Light

I'm building a geocache container. To make it look authentic, I'd like the red light to flash every few seconds. Unfortunately, I don't have any "Nuts & Volts" electronics skills — what type of long-lasting battery and low-power LED light to use — and am hoping to get some expert advice.

Search online for "firefly circuit 7555" and you will find what you want (or "unijunction transistor flasher" for the old fashioned method). The LED should be a high brightness type with diffused lens, at

All questions AND answers are submitted by *Nuts & Volts* readers and are intended to promote the exchange of ideas and provide assistance for solving technical problems. All submissions are subject to editing and will be published on a space available basis if deemed suitable by the publisher. Answers are submitted by readers and

NO GUARANTEES WHATSOEVER are made by the publisher. The implementation of any answer printed in this column may require varying degrees of technical experience and should only be attempted by qualified individuals.

Always use common sense and good judgment!

Send all questions and answers by email to forum@nutsvolts.com
or via the online form at www.nutsvolts.com/tech-forum

least 300 mcd. The LEDs above 1CD typically use a clear lens, so can't be seen off axis. As an alternate, if you only need it to run a couple of days, then just solder a L-36BSRD flashing LED inside a 9V battery snap. I make these up for cavers; they leave them behind like breadcrumbs to find their way back.

Bob Turner
Salamander Bay, Australia

[#11151 - November 2015]
Super Capacitor Comparison

I need some help understanding what's going on inside a super capacitor. I did some experiments using regular capacitors as a backup power supply to a real time clock chip. Mathematically, the amount of time an electrolytic capacitor (1500, 2200, and 4700 μ F) would power the chip became predictable once I came up with a formula. However, when I connected a super capacitor, the math broke down.

The real time clock should have exhausted the stored power in the super capacitor after exactly three days. Instead, it is still maintaining the correct time after four months! Clearly, something is physically different about a super capacitor. It acts more like a battery than a capacitor.

Can someone explain to me how the chemistry of a super capacitor differs from electrolytic capacitors? Why does the amount of charge stored seem to far exceed the capacity indicated by its Farad value, under an extremely light load?

#1 "Super caps" are, indeed, electrolytic capacitors. Their construction typically uses tantalum (not aluminum) plates to obtain a large capacitance value in a relatively small package compared to standard aluminum electrolytics. Plus, their electrolyte formulation differs enough from aluminum capacitors to allow a

larger charge vs. size capability.

Because of their large charge capability, they are ideal for use in short-term backup applications where low currents (typically, less than 100 microamperes) are required (i.e., real time clock chips). However, they are *not* the same as a battery (i.e., lithium coin cell) since — like regular electrolytics — super caps do self-discharge over time and they can *not* deliver a large supply current (i.e., >1 mA) for longer than a couple of seconds.

Ken Simmons
Auburn, WA

#2 For any given capacity rating, an electrolytic capacitor should behave very much like a super capacitor, at least in a low current circuit like a RTC. However, super capacitors provide much higher capacity in a given package size. So much so, that they are rated in millifarads ($1\text{ mF} = 1,000\text{ }\mu\text{F}$) or Farads ($1\text{ F} = 1,000,000\text{ }\mu\text{F}$) rather than microfarads (μF).

Are you sure you took this into account when you did your calculation? If not, you may be off by a factor of 1,000 or one million. The other possibility is leakage current causing the electrolytic capacitor to drain faster than you expect. This presents itself as if there was a resistor connected across the capacitor, constantly draining away its charge.

The leakage current value is listed on the datasheet for the capacitor and must be taken into account when you're using it for long storage applications like a real time clock.

Mark Lewus
Denville, NJ

#3 Your calculations are probably correct. However, you may have used the worst case current from the datasheet; if you use the nominal current, it should be good. A couple

of months is the expected retention at room temperature. Just be careful if you use Schottky diodes in your circuit (to power the RTC from normal 5V). The leakage current in a poorly selected diode can exceed the standby current. (Schottky diodes are made with three doping levels, producing forward drops of 200, 300, and 400 mV; the 400 mV has the lowest leakage current. It will usually have an H in the part number.) A super capacitor is a electrolytic double layer capacitor (EDLC), and each electrode is coated in very fine carbon granules. The total surface area is about 1,000 times higher than just aluminium foil, with a much thinner dielectric; hence the increase in capacitance (and drop in maximum operating voltage). There is no actual oxide layer for the dielectric. An EDLC has charged layers of ions a few molecules thick instead, and charge and discharge just moves the ions back and forth across the layer.

Bob Turner
Salamander Bay, Australia

[#12153 - December 2015]
LiPo Smart Charger

I'm looking for a smart charger for LiPo cells that can run from a solar panel. I've heard that the combination is incompatible because of fluctuations in output from the solar panel. Is this true? If so, is a workaround a larger panel?

There are a lot of solar to li-ion battery charger ICs. You might get one on a breakout board, like the following: www.ladyada.net/make/solarlipo

I made a small solar powered solar tracker that has a similar IC, and charges two li-ion batteries to about 8.0 volts. It works well. It's been in operation over a year now.

John McCullough
La Habra, CA

■ NEW PRODUCTS

Continued from page 21

fuzz can be added pre or post of the filter section, or be removed entirely from the circuit. The Tone knob adjusts between a darker and brighter fuzz, while the Bias knob takes the fuzz from "normal" to a "dying battery" sound. With its EXP input, the player can use an expression pedal to control and sweep the wah, and talking pedal filters in real time for an array of especially expressive sounds.

The Cock Fight comes equipped with a standard EHX 9.6DC-200mA PSU and is shipping now with a suggested retail price of \$148.90.

For more information, contact:
Electro-Harmonix
www.ehx.com

LOW POWER MCUs EXTEND BATTERY LIFE

Microchip Technology, Inc., has announced the expansion of its Low Power PIC® microcontroller (MCU) portfolio. Features of the new PIC24F "GB6" family include up to 1 MB of Flash memory with Error Correction Code (ECC) and 32 KB of RAM, making it their first 16-bit MCU to offer such a large memory size. Also featuring dual-partition Flash with live update capability,

these devices can hold two independent software applications, permitting the simultaneous programming of one partition while executing application code from the other. This powerful combination of features makes the PIC24F "GB6" family ideal for industrial, computer, medical/fitness, and portable applications that require a long battery life, and data transfer and storage without the need of external memory such as electricity metering, HVAC control, fingerprint scanners, and gaming.

The PIC24F GB6 family is supported by Microchip's standard suite of development tools. The new PIC24FJ1024GB610 Plug-In Module (part #MA240023, \$25) is now available for the Explorer 16 Development Board (part #DM240001, \$129.99). Pricing starts at \$1.74 each, in high volumes. Product variants are available in 64-pin, 100-pin and 121-pin packages, with Flash memory ranging from 128 KB to 1 MB.



For more information, contact:
Microchip
www.microchip.com

ADvertiser INDEX

Find your favorite
advertisers here!

3D PRINTERS/CNC

Maker's Tool Works 13
Probotix 29

CHARGERS

Hitec 3

AMATEUR RADIO / TV

National RF 21
Palstar 14

CIRCUIT BOARDS

AP Circuits 22
Circuit Gizmos 21
ExpressPCB 63
Front Panel Express LLC 50
Googilum 19
PCB Shopper 34
Saelig Co. Inc. 23

COMPONENTS

All Electronics Corp. 61
Electronix Express 34
Parts Express 12

DESIGN/ENGINEERING/ REPAIR SERVICES

Cleveland Institute of Electronics 22
ExpressPCB 63
Front Panel Express LLC 50
National RF 21

DEV PLATFORMS

Anaren Back Cover

EDUCATION

Cleveland Institute of Electronics 22
Command Productions 7
NKC Electronics 21
Poscope 14

ENCLOSURES

Front Panel Express LLC 50

KITS & PLANS

National RF 21
NKC Electronics 21
Qkits 21

MICROCONTROLLERS / I/O BOARDS

Ion Motion Control 12
M.E. Labs 51
Technologic Systems 5

MISC./SURPLUS

All Electronics Corp. 61
TEquipment 2

MOTORS / MOTOR CONTROL

Ion Motion Control 12
Qkits 21
Robot Power 21
The Robot Market Place 6

RF TRANSMITTERS/ RECEIVERS

Lemos International 43
National RF 21

ROBOTICS

Cleveland Institute of Electronics 22
Electronix Express 34
Hitec 3
Ion Motion Control 12
Robot Power 21
The Robot Market Place 6

TEST EQUIPMENT

NKC Electronics 21
Pico Technology 67
Poscope 14
Saelig Co. Inc. 23
TEquipment 2

TOOLS

Maker's Tool Works 13
PanaVise 50
Parts Express 12
Poscope 14
TEquipment 2

WIRE, CABLE AND CONNECTORS

All Electronics Corp. 61

WIRELESS PRODUCTS

Anaren Back Cover
Lemos International 43
Technologic Systems 5

All Electronics Corp. 61
Anaren Back Cover
AP Circuits 22
Circuit Gizmos 21
Cleveland Institute of Electronics 22
Command Productions 7
Electronix Express 34
ExpressPCB 63
Front Panel Express LLC 50
Googilum 19
Hitec 3
Ion Motion Control 12
Lemos International 43
Maker's Tool Works 13
M.E. Labs 51
National RF 21
NKC Electronics 21
Palstar 14
PanaVise 50
Parts Express 12
PCB Shopper 34
Pico Technology 67
Poscope 14
Probotix 29
Qkits 21
Robot Power 21
Saelig Co. Inc. 23
Technologic Systems 5
TEquipment 2
The Robot Market Place 6



7 functions of PicoScope:

1. Oscilloscope
2. Spectrum analyzer
3. Function generator
4. AWG
5. Logic analyzer
6. Serial protocol analyzer
7. Automatic waveform test

8 CHANNEL PC OSCILLOSCOPE



For just \$2300

- High resolution
- USB powered
- Deep memory

INCLUDES AUTOMATIC MEASUREMENTS, SPECTRUM ANALYZER, SDK, ADVANCED TRIGGERS, COLOR PERSISTENCE, SERIAL DECODING (CAN, LIN, RS232, I²C, I²S, FLEXRAY, SPI), MASKS, MATH CHANNELS, ALL AS STANDARD, WITH FREE UPDATES

12 bit • 20 MHz • 80 MS/s • 256 MS buffer • 14 bit AWG

Call Us At: 1-800-591-2796 www.picotech.com

COMING TO CES 2016



If you thought our original AIR for WICED BLE module and *Atmosphere*, online development platform were amazing – just imagine the fun you'll have with our:

- extended family of AIR for WICED modules
- totally modular, hardware development tools
- *Atmosphere* 2.0 platform
- and all-new *Stratosphere* cloud platform

Register now to be part of the official launch!



Anaren®

What'll we think of next?®